

**Best Available  
Copy  
for all Pictures**

AD-770 636

DIGITAL HARDWARE FOR A HYBRID OPTICAL/  
DIGITAL COMPUTER INTERFACE

David Casasent, et al

Carnegie-Mellon University

Prepared for:

Air Force Office of Scientific Research  
Advanced Research Projects Agency

September 1973

DISTRIBUTED BY:

**NTIS**

National Technical Information Service  
U. S. DEPARTMENT OF COMMERCE  
5285 Port Royal Road, Springfield Va. 22151

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER <b>AFOSR - TR - 73 - 2074</b>	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  DIGITAL HARDWARE FOR A HYBRID OPTICAL/DIGITAL COMPUTER INTERFACE		5. TYPE OF REPORT & PERIOD COVERED  Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)  David Casasent Warren Sterling		8. CONTRACT OR GRANT NUMBER(s)  F44620-73-C-0074
9. PERFORMING ORGANIZATION NAME AND ADDRESS  Carnegie-Mellon University Department of Computer Sciences Pittsburgh, Pennsylvania 15215		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS  61101E AO 2466
11. CONTROLLING OFFICE NAME AND ADDRESS  Advanced Research Projects Agency 1400 Wilson Blvd Arlington, Virginia 22209		12. REPORT DATE  September 1973
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)  Air Force Office of Scientific Research (NM) 1400 Wilson Blvd Arlington, Virginia 22209		13. NUMBER OF PAGES  63
		15. SECURITY CLASS. (of this report)  UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Reproduced by NATIONAL TECHNICAL INFORMATION SERVICE U S Department of Commerce Springfield VA 22151		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  An optical/digital interface for hybrid optical/digital processor has been designed and implemented at the register transfer level. An overall view of the hybrid processor is given followed by a discussion of the processing requirements of the interface. The interface hardware is then presented in depth with a short introduction to the register transfer level of design and associated hardware components. Finally, experimental results achieved with the hybrid processor are included.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DIGITAL HARDWARE FOR A HYBRID  
OPTICAL/DIGITAL COMPUTER INTERFACE

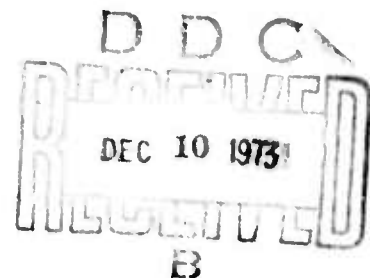
David Casasent  
Warren Sterling

September, 1973

Departments of Computer Science  
and Electrical Engineering  
Carnegie-Mellon University  
Pittsburgh, Pa. 15213

The research reported here was partially supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-73-C-0074) and monitored by the Air Force Office of Scientific Research.

Approved for public release;



## ABSTRACT

An optical/digital interface for a hybrid optical/digital processor has been designed and implemented at the register transfer level. An overall view of the hybrid processor is given followed by a discussion of the processing requirements of the interface. The interface hardware is then presented in depth with a short introduction to the register transfer level of design and associated hardware components. Finally, experimental results achieved with the hybrid processor are included.

## TABLE OF CONTENTS

1. INTRODUCTION .....	1
2. THE HYBRID OPTICAL/DIGITAL PROCESSOR.....	2
2.1. Basic Optical Data Processing Operations	
2.2. Input Modulator	
2.3. Optical/Digital Interface Functions	
2.4. Optical/digital Interface	
3. OPTICAL/DIGITAL INTERFACE SPECIFICATIONS.....	8
3.1. Typical Data Fonts	
3.2. Performance Requirements	
3.3. Theory of Operation	
3.4. Vidicon Interface	
3.5. Sync Separator	
4. RTM LEVEL INTERFACE DESIGN.....	22
4.1. Register Transfer Modules (RTM's)	
4.2. Video Processor and Digital Interface Design	
4.3. PDP-11 Interface Control	
5. EXPERIMENTAL RESULTS .....	47
6. SUMMARY .....	51
7. ACKNOWLEDGMENTS .....	52
8. APPENDIX .....	55

## FIGURE TITLES

- Figure 1. The hybrid optical/digital processor system. For simplicity, only the data paths for the output plane ( $P_2$ ) vidicon are shown. Paths for the Fourier transform plane ( $P_1$ ) vidicon are identical. The  $P_1$  plane is viewed via a beam splitter.
- Figure 2. Basic optical data processing system lens and planes.
- Figure 3. Block diagram of the optical/digital interface.
- Figure 4. Typical optical data processing input images and resultant outputs.
- 4(a). Phased array antenna data.
  - 4(b). Fourier transform of 4(a).
  - 4(c). Aerial photograph used as input image for scene correlation.
  - 4(d). Portion of 4(c) used for reference spatial filter.
  - 4(e). Optical correlation of 4(c) and 4(d).
  - 4(f). Input plane text used for text correlation.
  - 4(g). Optical correlation of 'HIGH' with 4(f).
- Figure 5. Standard 525 line 2:1 interlace vidicon scan.
- Figure 6. Video signals for standard 2:1 interlace 525 line vidicons.
- 6(a). Composite video format.
  - 6(b). Horizontal blanking pulses.
  - 6(c). Vertical blanking pulses.
- Figure 7. Binary digitization for one horizontal line of video signal with a resolution of 10 digital elements per line.
- Figure 8. The area covered by a typical digital picture element from a picture of resolution  $60 \times 60$ .
- Figure 9. Examples of binary digital images of two different resolutions.
- Figure 10. Diagram of vidicon interface.
- Figure 11. The programmable clock. The output is the SHIFT H signal with period  $T_c$ .
- Figure 12. The video thresholder. All resistor values are ohms and all capacitor values are microfarads unless specified otherwise.
- Figure 13. The 100 ns delay.
- Figure 14. Details of shift register and buffer.
- Figure 15. Photograph of vidicon interface card.

- Figure 16. Sync separator circuit. All resistor values are ohms and all capacitor values are microfarads unless specified otherwise.
- Figure 17. DM(general purpose arithmetic unit) control inputs, registers, and boolean outputs.
- Figure 18. Flow diagram of video processor and digital interface.
- Figure 19. Main processor RTM design.
- Figure 20. RTM Subroutine INTERFACE.
- Figure 21. Kmacro(DISPLAY)
- Figure 22. Kmacro(ACCEPT VIDEO WORD).
- Figure 23. Kmacro(NEWLINE).
- Figure 24. Examples of Ke evoke line connections.
- Figure 25. The Kwait module
- Figure 26. DR11-A digital control.
- Figure 27. DR11-B digital control.
- Figure 28. Block diagram of DR11-B control signal connections.
- Figure 29. Photograph of digital portion of the hybrid processor. From left to right: CRT display, TV monitor, rack mounted optical/digital interface, PDP-11 computer (in floor rack).
- Figure 30. CRT display image demonstrating digital resolution of  $64 \times 240$  picture elements for one on-line TV frame.
- Figure 31. Binary digital images of the optical processor output planes shown in Figure 4. Ignore vertical lines at sides of the pictures.
- 31(a). Off-axis spectral components for phased array radar processing output of 4(b). Resolution:  $64 \times 240$ .
  - 31(b). Correlation peak of scene correlation output of 4(e). Resolution:  $32 \times 30$ .
  - 31(c). Correlation peak of text correlation output of 4(g). Resolution:  $64 \times 120$ .



## 1. INTRODUCTION

This report describes the design and construction of digital hardware to be used in a novel hybrid optical/digital processor. In particular, the details of an interface between an on-line coherent optical processor and a PDP-11 computer will be presented. This optical/digital interface was designed primarily at the register transfer (RT) level. At this level, the components, or design primitives, are: registers, bytes or words of data (i.e., a defined set of bits); the transfer of such data between registers; and various logical and arithmetic operations upon this data. Table 1 summarizes the three logic design levels [1]. From this table the RT level is seen to be the next higher level of complexity above the conventional switching circuit level.

<u>LEVEL</u>	<u>TYPICAL CIRCUITS</u>	<u>TYPICAL COMPONENTS</u>
Register Transfer	Arithmetic Unit	Registers Transfers Controls Operators (+,-,etc.)
Switching (sequential)	Counters Function Generator Controls	Flip-flops Delay One-shot
Switching (combinatorial)	Encoders Decoders Bit operators Iterative networks	AND, OR NOT NAND, NOR

Table 1. Logic Design Levels

The optical/digital interface was implemented using register transfer modules (RTM's), a commercial product developed at Carnegie-Mellon University and distributed by Digital Equipment Corporation [2]. A register transfer

level design was chosen to demonstrate that a real-time system with high data rate requirements could be designed in a relatively short time and implemented with relatively few register transfer level components, when compared to the number of components required at the conventional switching circuit level. The final results, a processor which has been operational since January, 1973, support these claims.

Besides connecting the optical and digital systems, the interface drives a CRT display using positional data supplied by the PDP-11. In the future, it will also be used to interface the PDP-11 to various other peripherals of the hybrid processor, such as video storage devices.

## 2. THE HYBRID OPTICAL/DIGITAL COMPUTER

### 2.1. Basic Optical Data Processing Operations

To understand the constraints imposed on the optical/digital interface, the entire hybrid processor must be understood. Figure 1 shows a simplified block diagram of the hybrid processor. This system is described in more detail in reference [3]. The top section of the figure consists of a conventional coherent optical data processing system, shown separately in Figure 2.  $P_0$  is the input plane and  $L_1$  is a spherical lens. With a transparency in  $P_0$  illuminated by parallel monochromatic coherent light, the two-dimensional optical Fourier transform of the input image in  $P_0$  is formed in the Fourier transform plane  $P_1$  [4]. The Fourier transform can be imaged onto vidicon 1 in Figure 1 as shown. When a reference beam is combined with this Fourier transform light distribution, a hologram is formed [5] and the resulting distribution imaged on vidicon 1 contains both the amplitude and phase of all spectral components of the Fourier transform. Without the reference beam, the

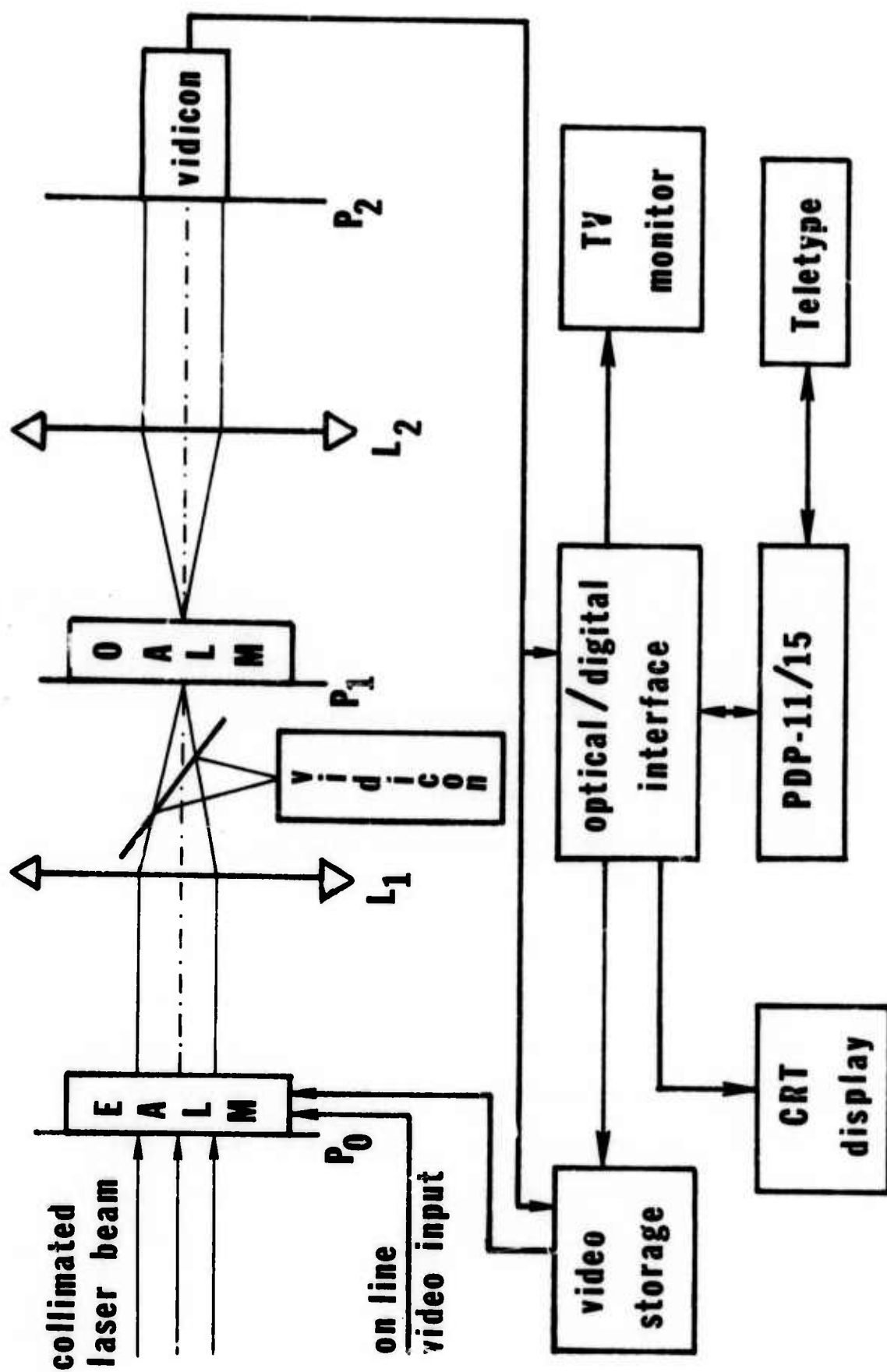


Figure 1. The hybrid optical/digital processor system. For simplicity, only the data paths for the output plane ( $P_2$ ) vidicon are shown. Paths for the Fourier transform plane ( $P_1$ ) vidicon are identical. The  $P_1$  plane is viewed via a beam splitter.

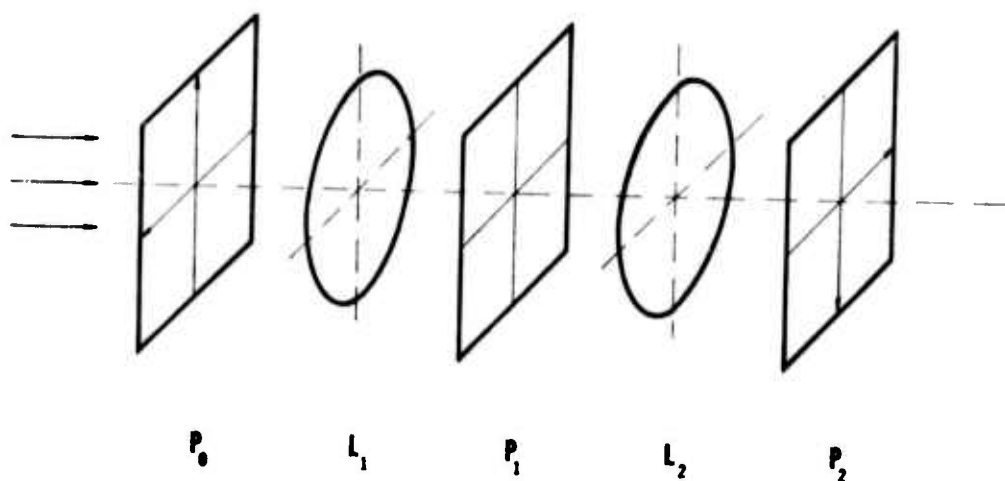


Figure 2. Basic optical data processing system lens and planes.

power spectrum of the input image is formed and the intensity of the various Fourier orders can be used to extract information concerning the input image by spectral analysis.

When an optically addressed light modulator (OALM in Figure 1) such as film, a thermoplastic, liquid crystal, photodichroic crystal, etc., is placed in plane  $P_1$ , and the transform of the input image is combined with a reference beam in  $P_1$ , a hologram of the input image can be recorded. When a new input image is placed in  $P_0$  and  $L_2$  is placed one focal length from  $P_1$  and  $P_2$ , the light distribution in  $P_2$  is the correlation of the original input image (now stored as a reference in  $P_1$ ) and the new input image in  $P_0$  [6].

With other patterns placed in  $P_1$  the frequency content of the input image can be altered. Such processing is generally referred to as spatial filtering. Perhaps the simplest example of the technique is to place a small opaque stop at the center of the  $P_1$  plane (the center of this plane corresponds to the dc term of the Fourier transform). This effectively passes only the high frequency components of the input image and the resultant reconstructed image appears differentiated [7]. These edge enhancement and image enhancement techniques which previously have been used to increase the quality of imagery are also applicable to computer vision and Artificial Intelligence applications.

## 2.2. Input Modulator

While the power of these optical processing techniques is well known in the optics community, they have not seen wide usage because of the lack of a real time input modulator (plane  $P_0$ ) and a real time adaptive spatial filter (plane  $P_1$ ). Film transparencies have typically been used in planes  $P_0$  and  $P_1$ . Their long development times and the inability to update the data once it is recorded on the film are severe disadvantages of such a system.

A real time electronically addressed light modulator (EALM in Figure 1) capable of converting an input electrical signal such as a raster scanned TV image into a transparent image capable of spatially modulating coherent light has been developed at CMU. The device is extensively described in the literature [8], [9], [10]. It consists of a modulated scanning electron beam which deposits a charge pattern on an electro-optic target crystal. The charge pattern deposited produces a variable electric field across the crystal surface which is thus capable of spatially modulating the amplitude of a collimated laser beam point by point by the linear longitudinal electro-optic or Pockels effect [11].

1000 x 1000 point images with over 10 grey levels have been processed at TV rates of 30 images per second for a data rate of  $3 \times 10^8$  bits/sec. with this device.

Input data for the EALM of Figure 1 can be input video data, a TV frame from video store, or any pattern digitally stored or generated by computer. Any light distribution incident on the OALM (plane  $P_1$ ) can be stored on the OALM. When the Fourier transform of a generic filter pattern such as a ring or wedge is placed in the input plane  $P_0$ , the image of the filter can be stored on the OALM. The hybrid processor thus has a real time adaptive input plane and spatial filter plane. Complex two dimensional Fourier transforms and correlations on  $10^6$  element images, therefore, are possible in real time at TV rates.

### 2.3. Optical/Digital Interface Functions

The principle applications of the system to be considered are radar processing and correlation. Optical processing of radar signals is discussed in reference [12]. In this application a properly chosen radar signal format written in the input plane is Fourier transformed [13]. The resultant pattern on vidicon 1 in Figure 1, consists of an arrangement of bright spots of light. The location of these peaks contains the desired azimuth, range, and doppler data for the input in question.

In a coherent correlation configuration, the output pattern in  $P_2$  on vidicon 2 in Figure 1 contains correlation peaks whose position and number provide information on the degree of correlation of two images.

The two vidicons are the primary link between the optical processor and the digital computer. Numerous other applications of the system exist in which case the desired output information from the optical processor consists of the

location and number of light intensity peaks in the output plane,  $P_2$ . The role of the interface is to extract this data and transfer it to the computer for storage, display, or to be acted upon to control the entire hybrid processor. When the output plane contains a coordinate image, the computer must store the entire frame of information and be able to display it on command. These functions and interface operations are performed on line at TV rates and are under program or teletype control. Section 3.1 further amplifies these processing requirements.

#### 2.4. Optical/Digital Interface

Figure 3 shows a block diagram of the optical/digital interface shown in the center of Figure 1. The interface consists of three sections:

1. Vidicon Interface
2. Video Processor
3. Digital Interface and Control

The vidicon interface transforms the video signal into a bit pattern, the video processor forms the final digitized image to be transferred to the PDP-11 while the digital interface and control section accepts commands from the PDP-11 to control all parts of the system (CRT, video storage, video processor, EALM, OALM, etc.). Figure 3 shows the interconnections of these sections.

The interface and its RTM level design will be extensively described in the following chapters. The remaining section of Figure 1 is the digital portion which consists of a PDP-11/15 with 12K of core, two low speed general purpose interfaces (DR11-A) and a high speed direct memory access interface (DR11-B). The DR11-B is used for all data transfers from the optical/digital

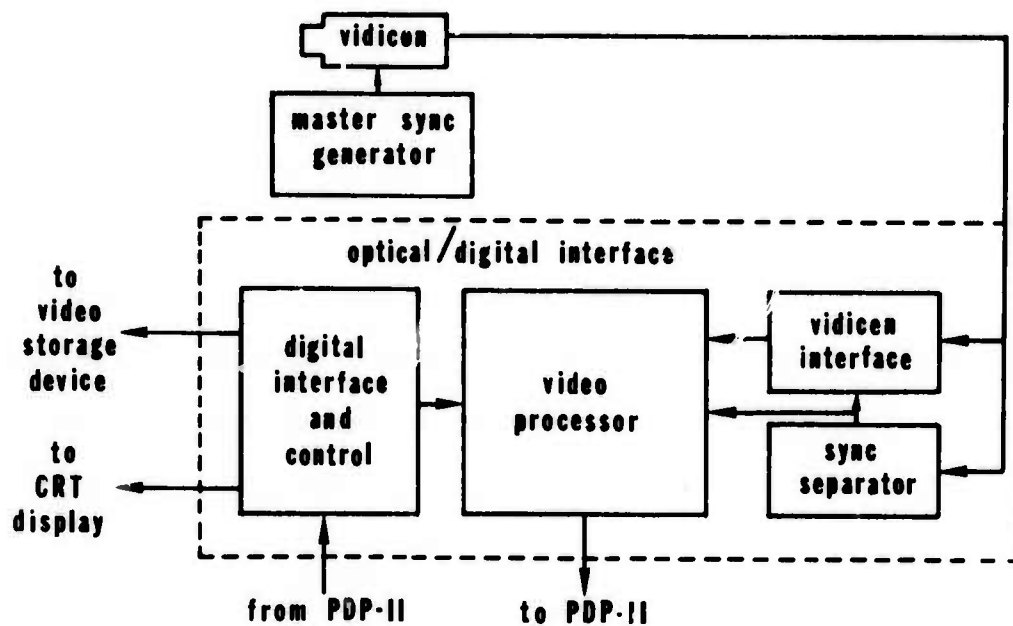


Figure 3. Block diagram of the optical/digital interface.

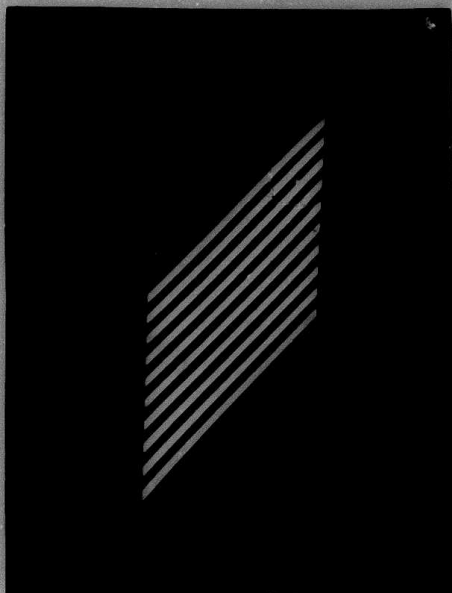
interface. A programming system to facilitate both conversational and stored program control of the system is currently being designed. A teletype and paper tape reader complete the digital portion of the hybrid processor.

### 3. OPTICAL/DIGITAL INTERFACE SPECIFICATIONS

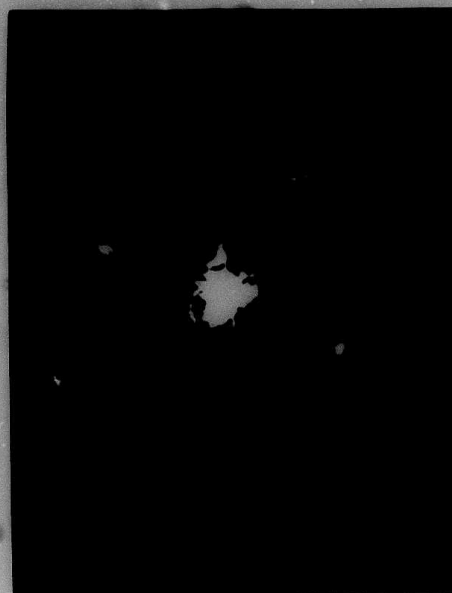
#### 3.1. Typical Data Fonts

As noted in Section 2.3, the analysis of the correlation and Fourier transform plane involves determining the presence and location of all intensity peaks (correlation points in the correlation plane or frequency components in





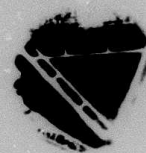
4(a). Phased array antenna data.



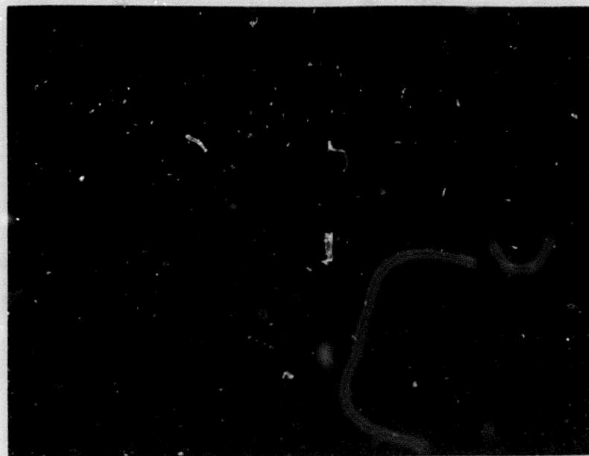
4(b). Fourier transform of 4(a).



4(c). Aerial photograph used as input image for scene correlation.



4(d). Portion of 4(c) used for reference spatial filter.



4(e). Optical correlation of 4(c) and 4(d).

**DANGER  
HIGH  
VOLTAGE**



4(f). Input plane text used for text correlation.

4(g). Optical correlation of 'HIGH' with 4(f).

Figure 4. Typical optical data processing input images and resultant outputs.

the Fourier transform plane). The operation is to be distinguished from the frequency analysis of a complex scene.

Figure 4 shows representative examples of three typical input images and the resultant outputs of the optical processor. The input pattern in Figure 4(a) is that associated with the data returns from a phased array antenna. The Fourier transform of this image, shown in Figure 4(b), consists of a central dc spot and two off-axis spectral components. The locations of these off-axis or first order components provides information on the targets' azimuth, range, and velocity [13] and are the quantities of interest to be extracted by the interface and digital computer.

Figures 4(c) and 4(d) are typical examples of scene correlation. With the image in Figure 4(c) placed in plane  $P_0$  of Figure 1 and a reference spatial filter of a portion of the image, Figure 4(d), placed in the spatial filter plane  $P_1$ , the output plane light distribution of Figure 4(e) results. This represents the cross correlation of the two images (reference and input). The intensity of the correlation peak in the output plane is a measure of the similarity (or correlation) between the two images. The location of the peak in the output plane represents the location of the triangular structure in the input image.

Figures 4(f) and 4(g) are an example of text correlation. The input text of Figure 4(f) is correlated against the spatial filter of the word 'HIGH' with the result shown in Figure 4(g). The location of the correlation spot corresponds to the location of the word 'HIGH' in the input image. Note the partial correlation of this word with the other words in the input image. In a complex situation with an entire page of text as the input image, many

correlation peaks occur. The interface must locate the position of each. In practice strong cross-correlations characterized by correlation peaks of slightly less intensity will occur at locations where the frequency content of the reference word or letter closely resembles that of another word or letter in the input image. In this case proper thresholding of the intensity of a "1" level in the output plane (i.e., rejection of unwanted cross-correlation peaks) is crucial for analysis. Reference [7] provides several examples of these and similar optical spatial filtering operations.

### 3.2. Performance Requirements

With TV frame rates, the computer must completely analyze one T frame before the next is written. The total processing time for digitization and analysis must not exceed 33 msec (1 TV frame time). A problem common to many processors is the collection of too much data. Both for speed considerations and in the interest of optimizing the system, it is imperative that the data transferred to the PDP-11 be minimized to be consistent with the accuracy necessary in the specific analysis being performed. This data compression phase of data analysis and collection is often ignored. Clearly this criteria changes from frame-to-frame and the extent of the data reduction performed by the video processor optimally should be under program control. Toward this end, the following digitization process is used:

1. The image is divided into a number of picture elements, the number of elements ranging from  $2 \times 2$  to  $128 \times 240$  with the resolution under program control. This range of resolution appears to be adequate for the processing situations analyzed thus far. Further data analysis is needed to generalize on the optimum output plane resolution for a general or even a dedicated hybrid optical processor.

2. Each element in the output image is given a value of "1" or "0" depending on whether or not the video signal within that element exceeds the programmable threshold voltage level. The capacity for 32 levels of intensity currently exists with much additional data analysis needed to determine the optimum number of quantized levels.

The processing rates require that a maximum of  $128 \times 240 = 3072$  bits be examined, all levels located, all statistical compilations performed, and all necessary system control functions initiated within 33 msec.

### 3.3. Theory of Operation

Planes  $P_1$  and  $P_2$  of the optical processor are continually scanned by standard 525 line 2:1 interlaced 30 frames/sec (60 fields/sec) vidicons. The automatic gain control on these units is disabled to allow them to record the true incident light levels. Figure 5 shows the raster scanning that results

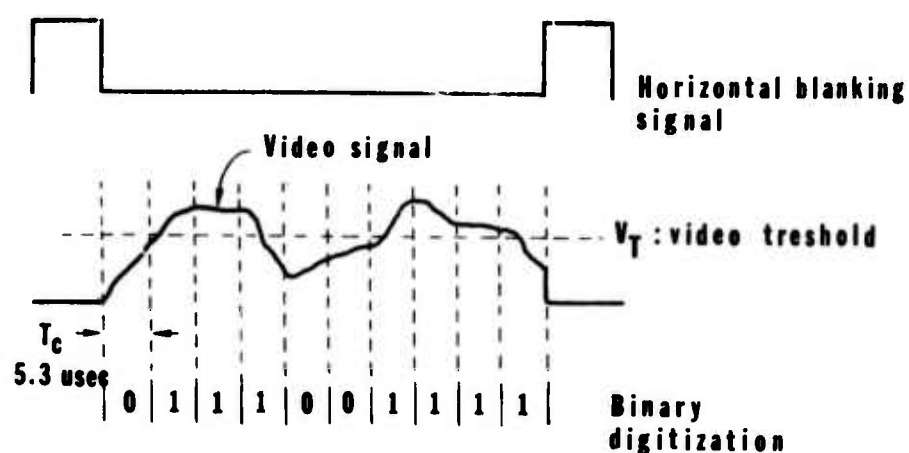
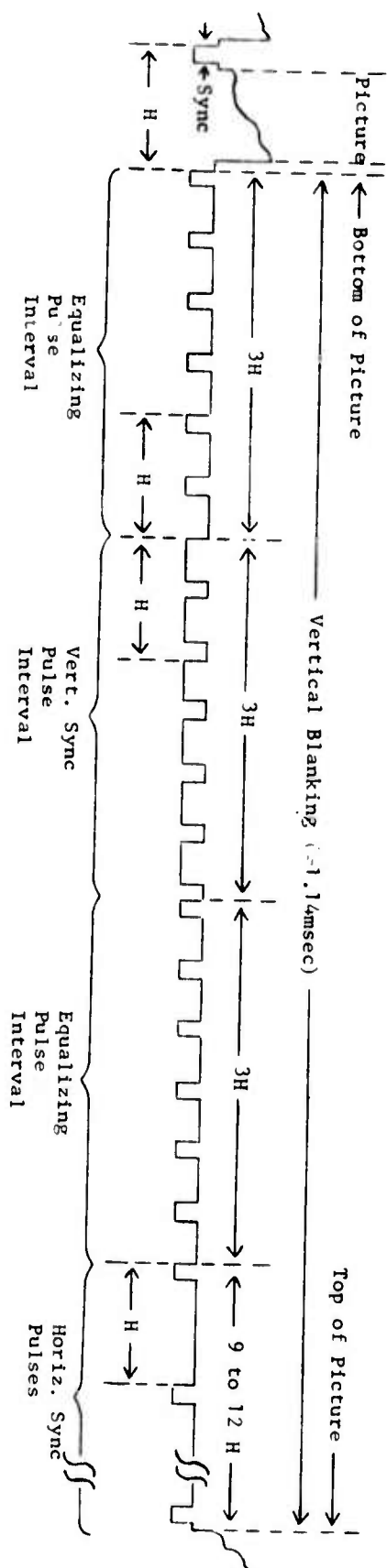
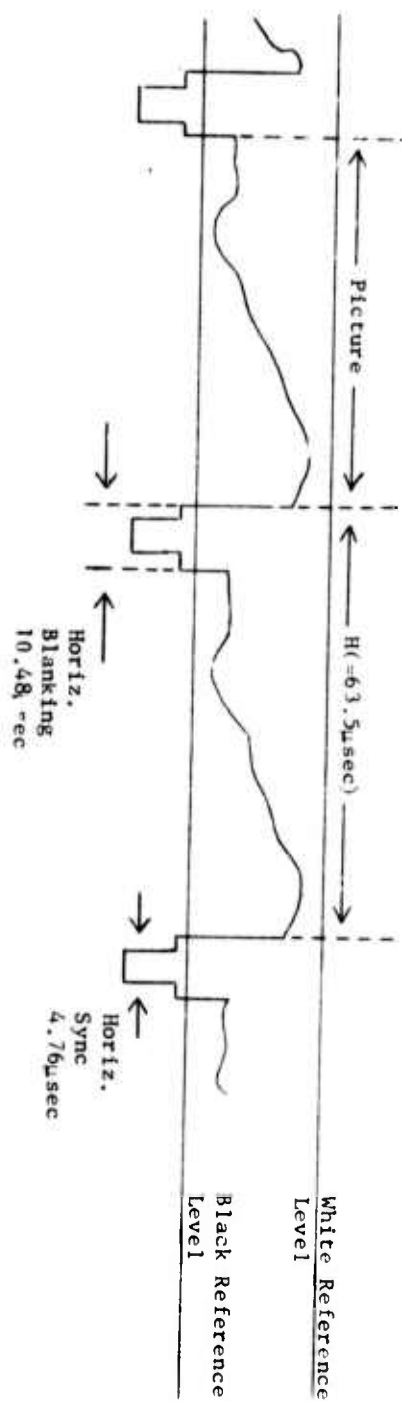
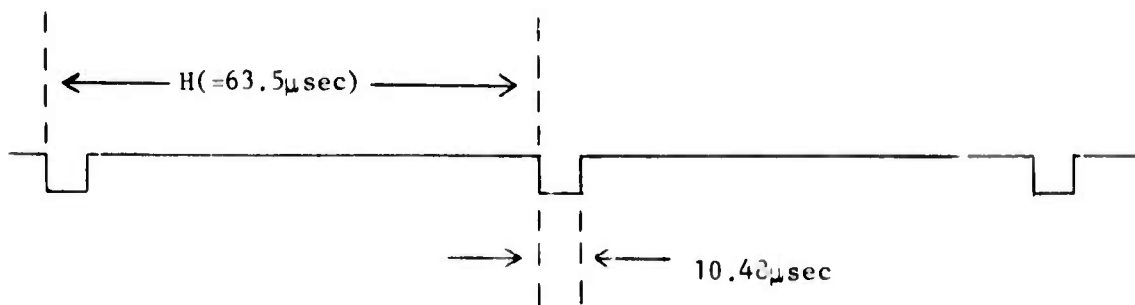


Figure 5. Standard 525 line 2:1 interlace vidicon scan.

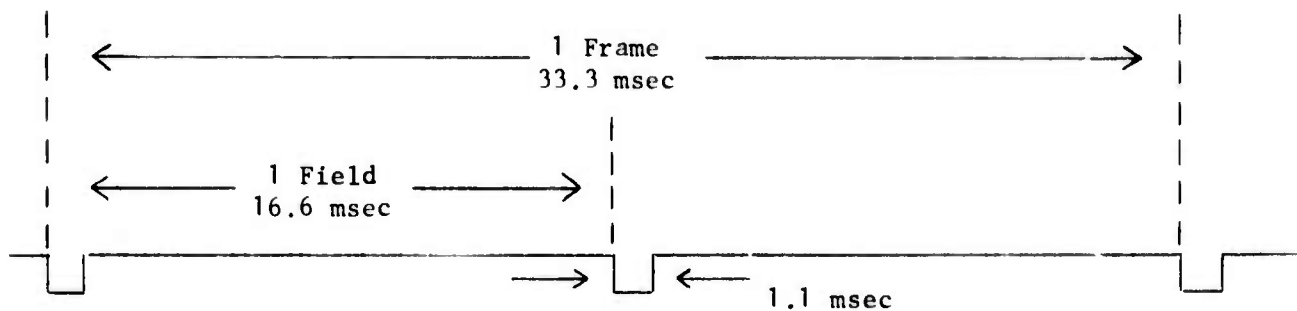




6(a). Composite Video Format



6(b). Horizontal Blanking Pulses



6(c). Vertical Blanking Pulses

Figure 6. Video signals for standard 2:1 interlace 525 line vidicons.

from a 2:1 interlaced scan. The odd field (lines 1,3,5,...) is written first, followed by the even field (lines 2,4,6,...). Figure 6 shows the various control signals associated with the video signal. These signals provide synchronization and timing for the scanned beam. They also provide all timing for the optical/digital interface, and, therefore, must be understood.

The standard vidicon output in Figure 6(a) is a composite of the video and sync signals. This composite signal is fed directly to the vidicon interface. The horizontal and vertical sync signals shown in Figure 6(a) normally drive the TV monitor. While they can be generated by a circuit internal to the vidicon, a master external sync generator was used here to facilitate synchronization of the various video operations: input video, vidicon scanners, TV

monitor, displays, etc. Besides sync pulses, blanking pulses as shown in Figures 6(b) and 6(c) are used by the TV monitor to suppress the write beam when video information is not being written on the phosphor. These blanking signals are used to drive the vidicon interface and video processor. They are generated from the composite video signals by conventional hardware.

Before initiating conversion to digital data, horizontal and vertical resolution values and a control word for the threshold voltage are fed to the optical/digital interface from the PDP-11. A start signal is then generated by the PDP-11 whereupon all operations are automatic, with all necessary timing signals derived from the blanking signals. Operation starts only at the beginning of a complete field. The interface then converts the optical image (represented by the video scan signal) to a binary digital image, that is, an image represented by a matrix of discrete picture elements, with each element having only two grey scale levels, black or white, represented by a 0 or 1, respectively.

Only one field of the video scan is used in the conversion process, rather than the entire frame, limiting resolution to  $262\frac{1}{2}$  lines, theoretically. As a practical limit, 240 lines was chosen as the maximum resolution to ensure that no lines falling within the vertical blanking period are digitized.

Figure 7 shows the digitization operation on a sample video signal entering the vidicon interface. As shown, each horizontal scan line is digitized according to the programmed horizontal resolution (for simplicity, a horizontal resolution of only 10 elements is shown in Figure 7). The video scan line for this case is divided into 10 equal elements of time  $T_c$ , each of 5.3  $\mu$ sec duration. The video level within each of these 10 elements is compared with the threshold voltage  $V_t$  selected.  $V_t$  can range from  $1.0V \geq V_t \geq 0.3V$  under program



control. Whenever the video signal anywhere within an element exceeds  $V_t$  a "1" value is assigned to that element, otherwise it receives a "0". The 10 bit word 0111001111 digitally represents the sampled video signal shown. This binary data is buffered and transferred in blocks of up to 16 bits to the video processor. For 64 element resolution, four data transfers of 16 bits each will occur. The detailed design of the vidicon interface is discussed in Section 3.4, only the operations it performs are created here.

The second section of the interface is the video processor. This combines the data from successive horizontal lines into the final digital picture elements according to the programmed vertical resolution. At the full 240 element vertical resolution, no horizontal lines are combined. With 120 element vertical resolution, each successive pair of horizontal lines is combined into one row of elements, etc.

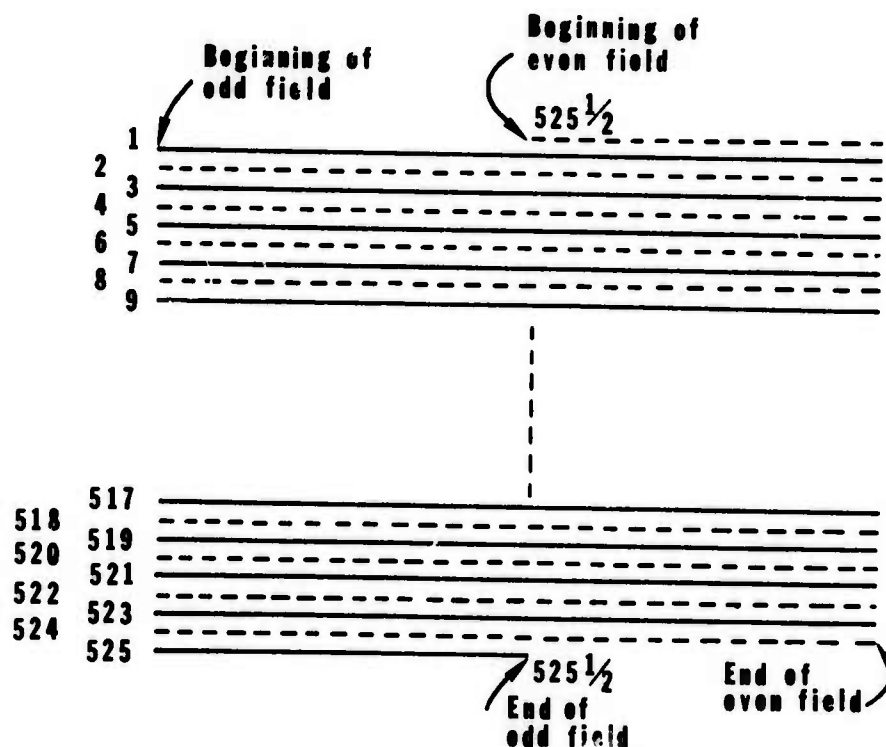


Figure 7. Binary digitization for one horizontal line of video signal with a resolution of 10 digital elements per line.

Figure 8 shows the area covered by one digital picture element in a TV image digitized into  $60 \times 60$  elements. Data from successive horizontal scan lines (19,21,23,25) is combined such that each digital picture element is a "1" if the video signal level anywhere within the area covered by the element exceeds the threshold voltage level. Figure 9 illustrates this resolution quantization for a  $4 \times 4$  and a  $10 \times 10$  element resolution image.

Other techniques for determining the value (1,0) of an element could include integration of the video signal within the area of the picture element followed by a thresholding decision; however, the compression algorithm chosen will detect the presence of a specified video signal level within the element regardless of its duration. This is the preferred method for a correlation and Fourier transform plane digitizer.

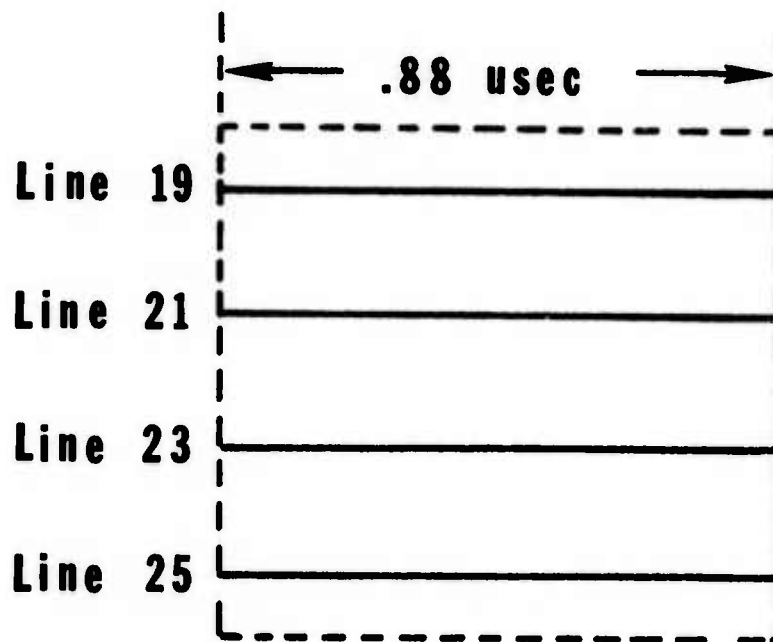


Figure 8. The area covered by a typical digital picture element from a picture of resolution  $60 \times 60$ .



**Image**

0	0	0	0
0	1	0	0
0	0	0	0
0	0	0	0

**4 x 4**

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

**10 x 10**

Figure 9. Examples of binary digital images of two different resolutions.

The output of the video processor is now a binary digital image. This image is transferred to the PDP-11 by either a DR11-A general purpose interface or a high speed direct memory access (DMA) DR11-B interface. At this writing, only the DR11-A has been used. The DR11-A has a limited horizontal resolution of 64 elements and thus the DR11-B is being implemented. The digital control for these interfaces is presented in Section 4.3.

It was noted earlier but is worth repeating that the entire process of

1. thresholding the video original,
2. forming the digital picture elements, and
3. transferring the entire image to the PDP-11

takes place within one vidicon field scan time of 15 msec. (Recall that two fields comprise a frame.) The process can be repeated for each ~~frame~~ or 30 times per second. Each digital image is thus formed from one field rather than one complete frame.

The device interface and control section of the optical/digital interface is also capable of driving a CRT display using positional data supplied by the PDP-11. It can be used in this fashion only when the video processor is not forming a digital image. This limitation is imposed because the device interface and the video processor share a common bus, as will be seen later; hence, they cannot operate in parallel. Their operation could be merged, particularly in the case of low resolution digital images; however, stringent time constraints would then be imposed on the video processor for high resolution (128 x 240) digital image generation. The two units are thus operated in exclusive time slots.

The CRT display technique is quite simple. No Z axis modulation of the CRT electron scanning beam is used. The PDP-11 calculates the coordinates of every picture element in the stored digital image that contains the value "1". These coordinates are then fed to the device interface together with a prefix code identifying the data word as either an X or Y coordinate. The interface decodes the data word and loads the coordinate value in either an X or Y axis D/A converter which then drives the scanning electron beam directly.

The display can be on-line, displaying each digital image as it is generated, or off-line, displaying an image other than the one just generated. The only limitation imposed on the number of images stored and displayed is the current 12K word memory capacity of the PDP-11.

The previous description of each of the optical/digital interface sections was purposely restricted to clearly show the operations each must perform. The detailed design of each of these sections will now be considered.

### 3.4. Vidicon Interface

The block diagram of the vidicon interface is shown in Figure 10. Figures 11 through 14 show the details of the blocks in Figure 10. Figure 15 is a photograph of the completed vidicon interface. The vidicon interface is contained on one double height DEC board and requires 16 IC packages. All interconnections in the entire optical/digital interface are wire wrapped.

The programmable clock shown in Figure 11 is driven by a 20 MHz oscillator constructed from two D-type flip-flops phase-locked to the blanking pulses. The horizontal length of a digital picture element or cell is a time denoted  $T_c$ . The period of the output clock pulse is  $T_c$  and can be programmed by parallel loading the counter with the frequency control word.

Figure 12 shows the video voltage thresholder circuit. The threshold voltage is set by the 5-bit threshold voltage control word which feeds an 8-bit D/A converter whose output is a 0 to 1 V level. The 710 IC differential comparator performs the actual comparison of the threshold voltage and the input video signal. The R-S flip-flop insures that a "1" output from the comparator will be held at the shift register input. The R-S flip-flop is set by the comparator and reset by the inverted shift pulse from the programmable clock of Figure 10.

The 100 nsec delay is generated by the RC integrator of Figure 13. The shift register and buffer into which the video thresholder shifts its output comparison levels is shown in Figure 14. Every  $T_c$  seconds (the horizontal length of an element) a new digital picture element bit is loaded into the shift



register. After every 16 bits, or at the end of a horizontal line (this is denoted by the occurrence of the RESET pulse), whichever occurs first, the 16 bit buffer is loaded. The "datain" flag is then set; this signals the video processor that a data word is available.

All signal designations have been made as self-explanatory as possible. The H or L following each signal denotes whether the signal is high (H) or low (L) when asserted.

### 3.5. Sync Separator

Figure 16 shows the sync separator circuit. This circuit accepts the composite video input signal from which the horizontal and vertical blanking signals needed for the interface timing are generated by integrators and monostables. All output levels are TTL compatible.

## 4. RTM LEVEL INTERFACE DESIGN

### 4.1. Register Transfer Modules (RTM's)

As previously noted, the video processor and digital interface sections of the optical/digital interface are constructed from existing modular computer elements, referred to as RTM's. Before detailing the remainder of the interface design, a brief explanation of RTM's is necessary. A more extensive description of these modules and their use is available in the literature [2], [14], [15], [16], [17].

An RTM system roughly consists of a set of data and control modules connected to a common 16 bit data bus. Each module can store and/or operate on one or more words (16 bits) or bytes (8 bits) of data. Data words are transferred between these modules by the data bus. Table 2 lists the five main

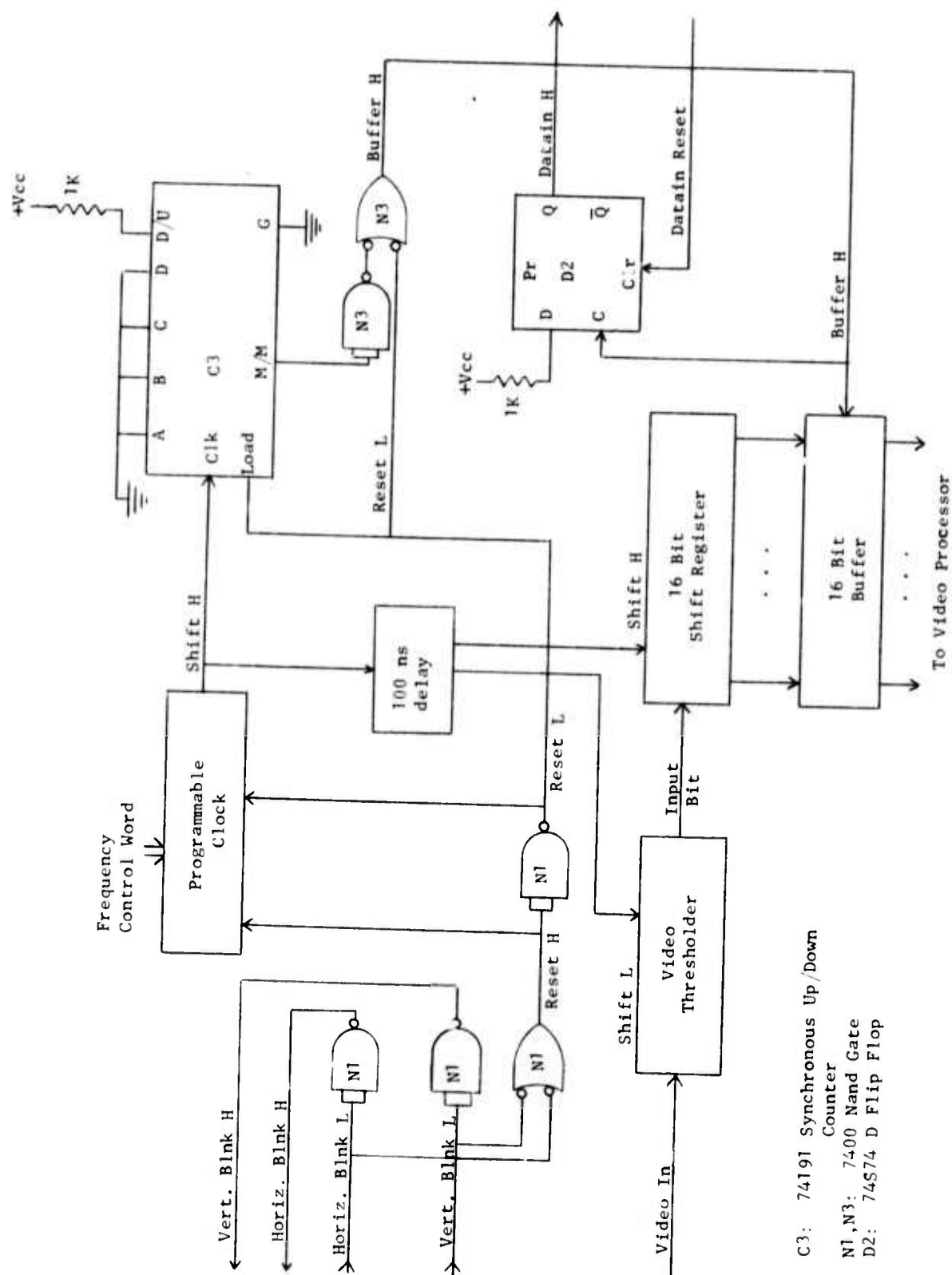
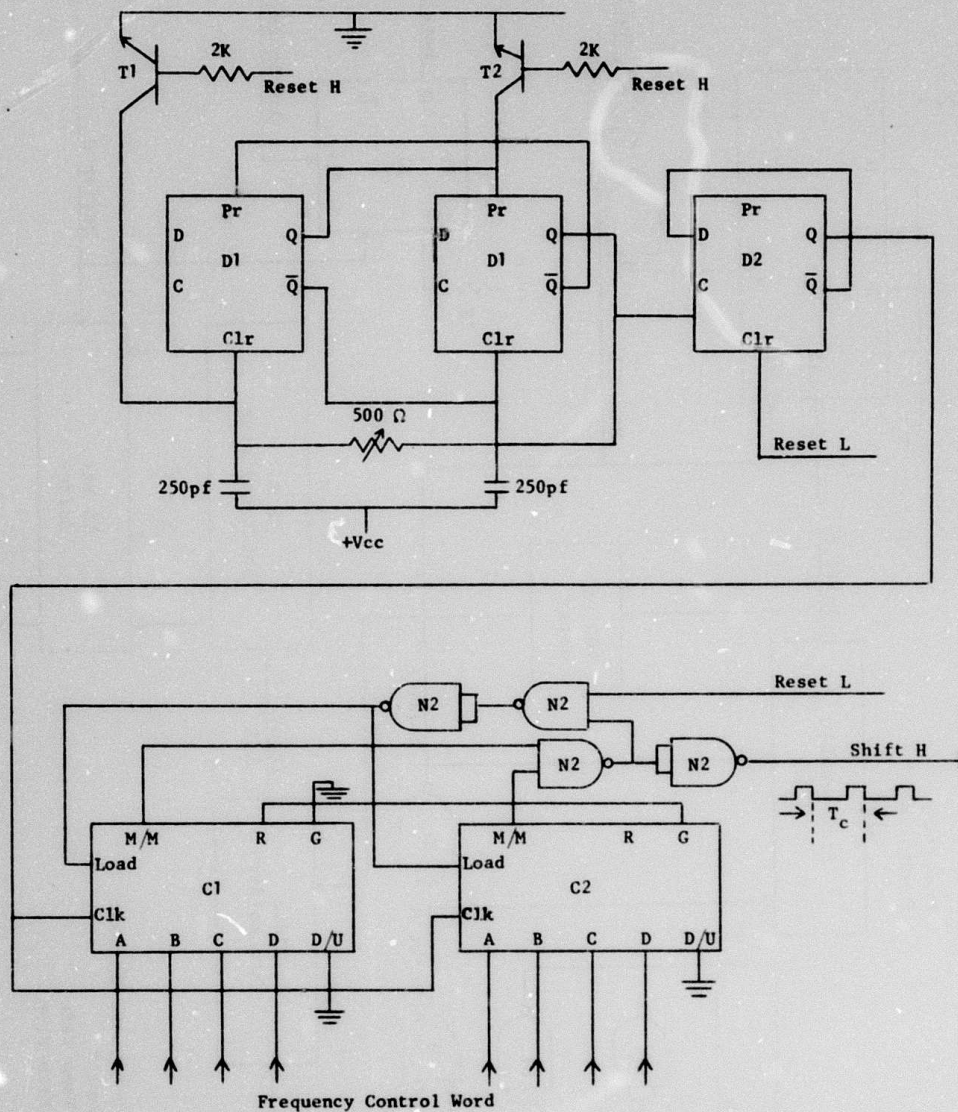


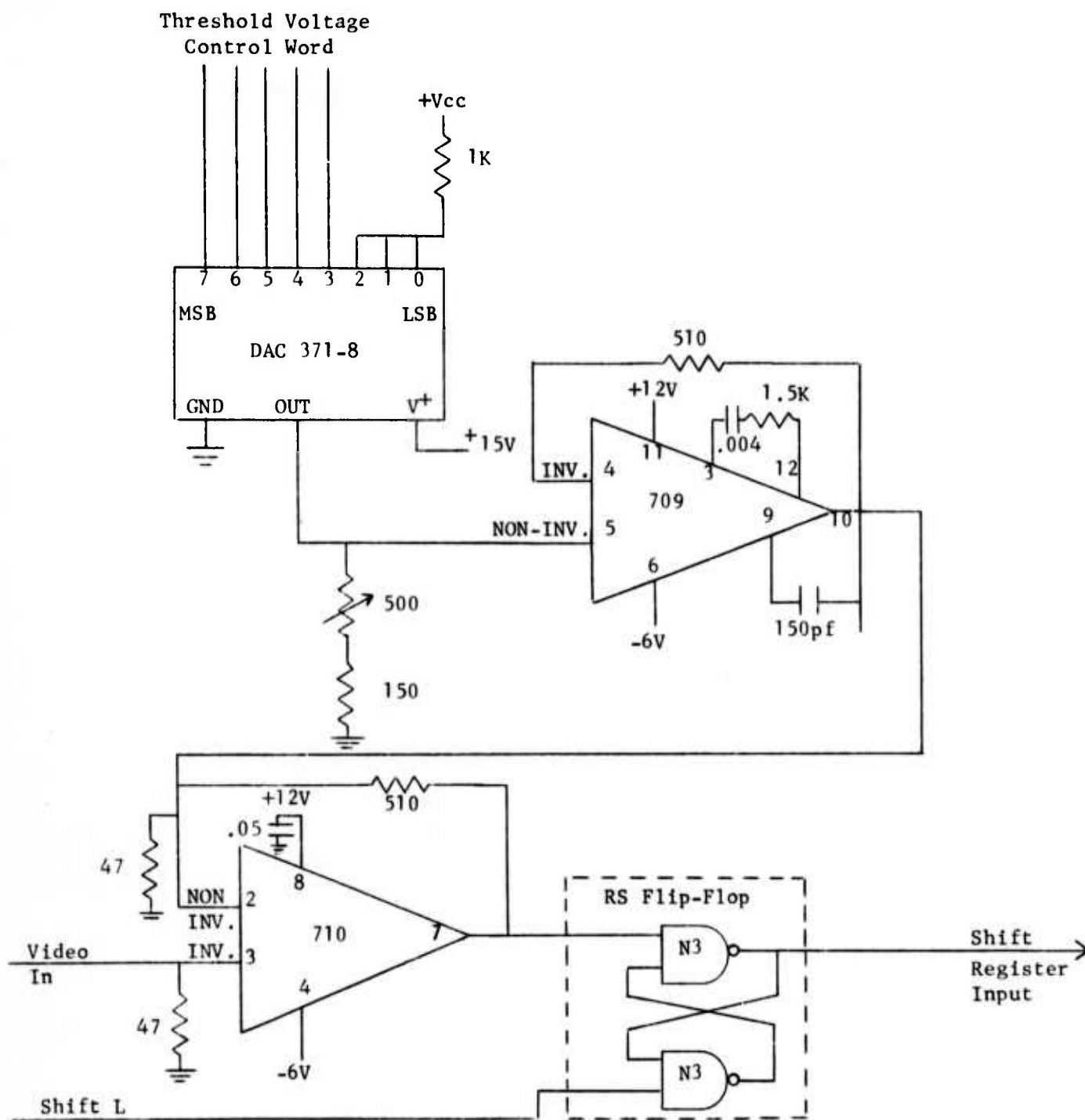
Figure 10. Diagram of Vidicon Interface



D1,D2: 74S74 D Flip Flops  
 C1,C2: 74191 Synchronous Up/Down Counters  
 N2: 7400 Nand Gates  
 T1,T2: 2N4124 TNX

Figure 11. The Programmable Clock. The output is the SHIFT H signal with period  $T_c$ .





DAC 371-8: 8 Bit D/A Converter  
 709: Operational Amplifier  
 710: Differential Comparator

Figure 12. The Video Thresholder. All resistor values are ohms and all capacitor values are microfarads unless specified otherwise.

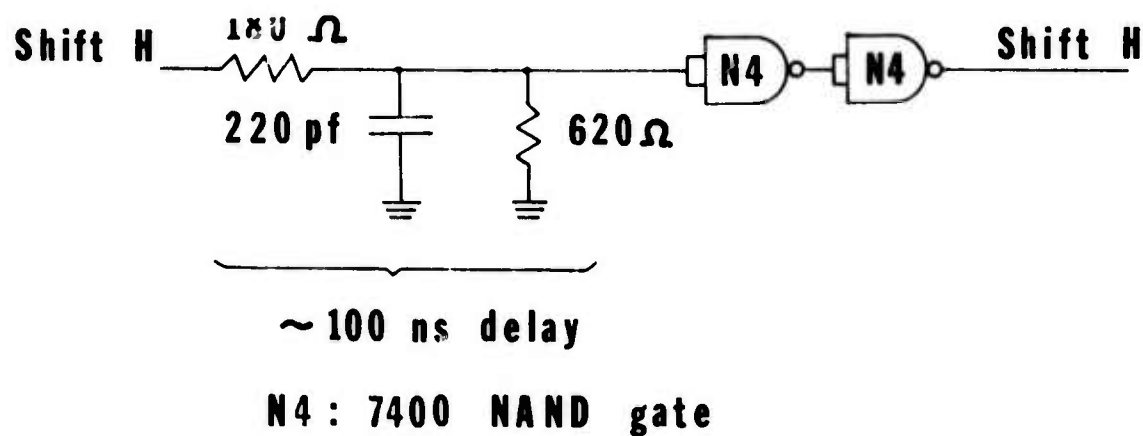
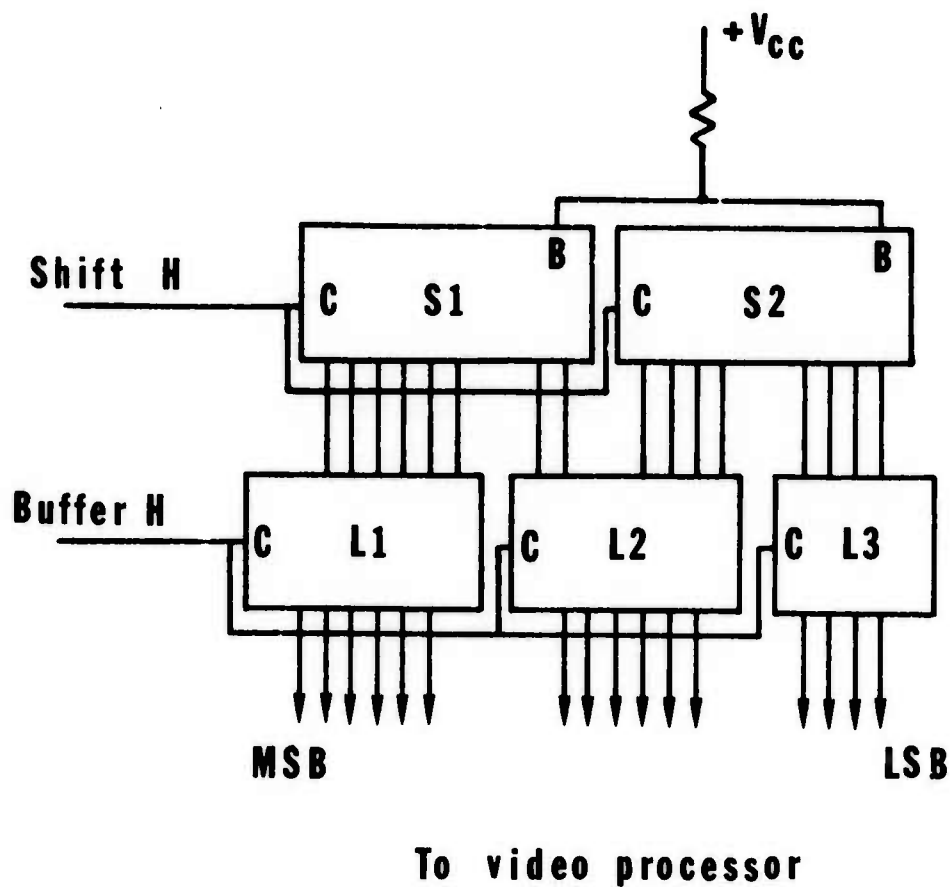


Figure 13. The 100 ns delay



S1, S2:	74164	8 bit shift register
L1, L2:	74174	6 bit latch
L3 :	74175	4 bit latch

Figure 14. Details of shift register and buffer.

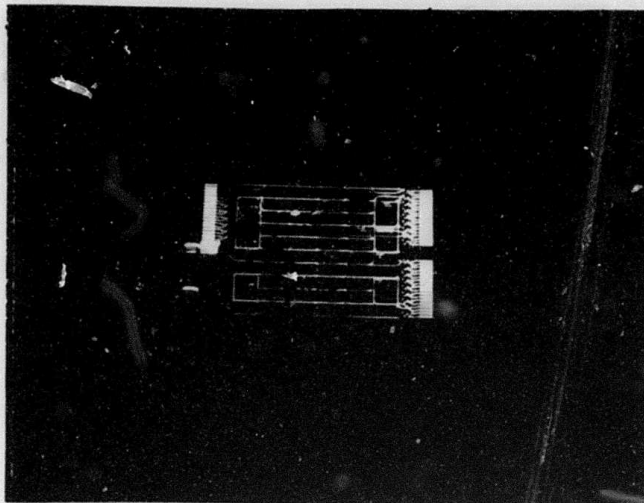


Figure 15. Photograph of Vidicon Interface Card

types of RTM's using the PMS notation of Bell and Newell [1]. The module examples referred to in Table 1 are explained more fully in Table 3 and the next section. Most modules have control inputs which are used to evoke the various logical operations and data transfers. Only one module is discussed in detail here as an example.

<u>TYPE</u>	<u>DESCRIPTION</u>	<u>EXAMPLE</u>
M (memory)	stores data	M(array; 256w)
D (data)	performs data operations (arithmetic, logical, decoding, etc.)	
DM (data-memory)	stores data and performs operations	DM(general purpose arithmetic unit)
T (transducer)	changes data to encode a given meaning in some new form (voltage to bit pattern, etc.)	T(video) T(D/A converter)
K (control)	evokes operations of other components in the system.	K(bus) Ke, Kb, Ksub

Table 2. Types of RTM Modules

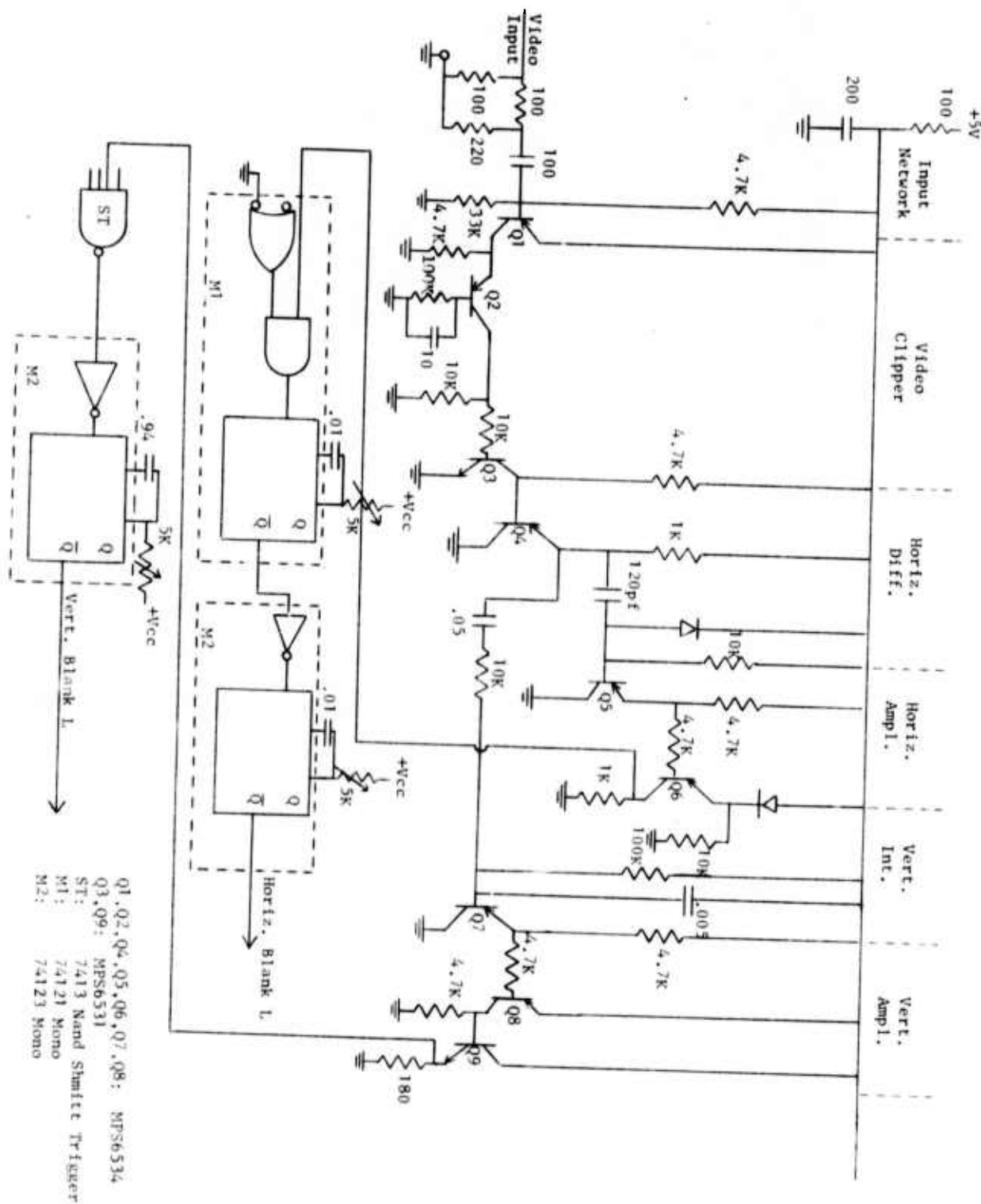


Figure 16. Sync separator circuit. All resistor values are ohms and all capacitor values are microfarads unless specified otherwise.

RTM control logic is designed with three basic control modules:

1. Ke: a module which initiates arithmetic operations, data transfers between registers, and memory read/write cycles.
2. Kb: a module which chooses a control path branch based on the value of a boolean flag.
3. Ksub: a module which transfers control to a closed set of RTM operations, and regains control upon their completion, in much the same manner as a software subroutine call.

These control modules will be illustrated fully by way of example in the next section.

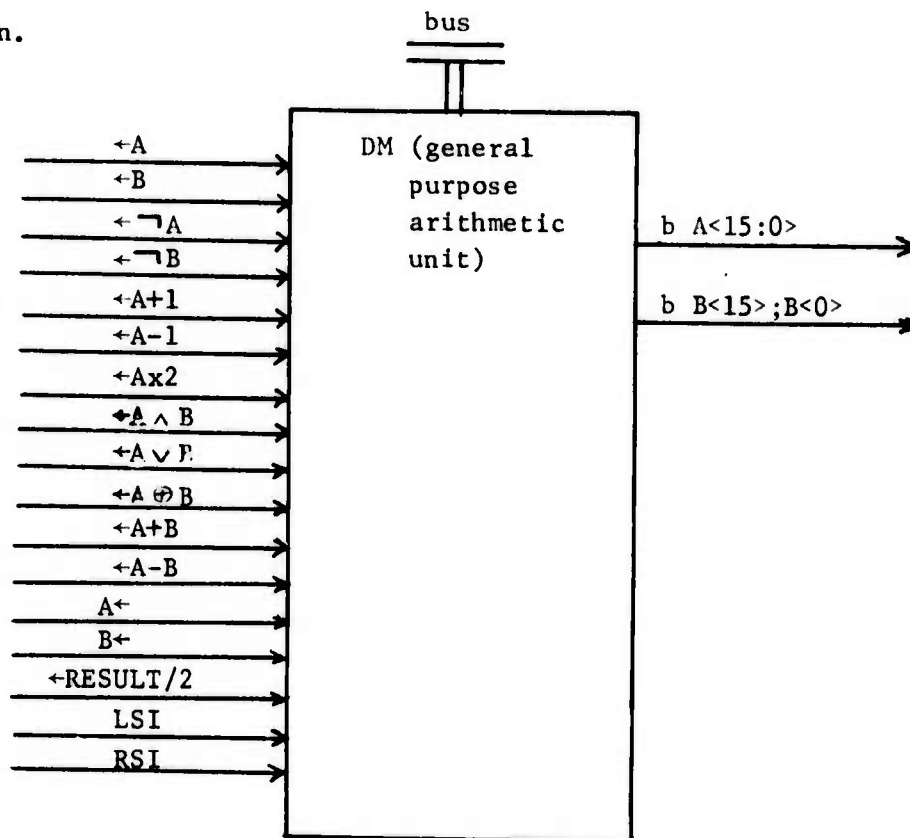


Figure 17. DM(general purpose arithmetic unit) control inputs, registers, and boolean outputs.

The general purpose arithmetic unit of Figure 17 is the major module in any RTM system. It can store two words of data in registers A and B, and perform 11 different operations on them. If the " $\leftarrow A+B$ " control line is asserted, the contents of the A and B registers are added and the result placed on the bus. If the " $B \leftarrow$ " control line is asserted, the contents of the bus will be loaded into the B register. The remaining control lines operate in a similar fashion. All results are in 2's complement form. Provisions for inputting end bits (LSI, RSI) for shifting operations are also included. All bits of the A register and the end bits of the B register are available as boolean outputs.

The control of an RTM system is completely asynchronous; there is no internal clock. A new operation begins only when the previous operation is complete. The time necessary for an operation is determined by the module performing that operation.

The use of RTM's combines the ease of software programming with the speed of a hardwired processor. All processor operations can be specified as algorithms. Given this specification, wiring lists can be automatically generated; the transition from an algorithm to the final hardwired processor is only one step. This approach also allows for easy alteration of the system since only backplane wiring is involved (assuming no custom-made modules are used). Another advantage of an RTM design is the possibility of sharing modules among processors. Since backplane wiring completely determines the processor, the modules themselves can be removed and used in another processor, thus realizing a significant cost reduction.

To illustrate the ease of processor designs at the RT level using these modules, it should be noted that this interface design was completed within two weeks (including the switching circuit design of the custom-made modules) by one person.

#### 4.2. Video Processor and Digital Interface Design

Figure 18 is a flow diagram of the complete video processor and digital interface. The processor is either waiting to start or looping through the actual processing section. Table 3 lists the actual modules used in the processor and defines all registers and flags associated with each module. The PMS notation is again used for the module names.

Figures 10 to 23 show the entire RTM system design. Figure 19 shows the main processor flow, and will be explained in detail to illustrate the uses of the Ke, Kb, Ksub, Kmacro, and Kwait control modules.

A Ke, or "evoke", module, initiates a register transfer operation via the bus. The first Ke in Figure 19 causes a zero to be loaded into register IN. The second Ke causes  $240_{10}$  to be loaded into register LINTOT, and sets flags "firstline" and "skipline" to 1. This Ke first asserts a control line to the M(4 word read only memory) module to initiate the loading of the constant register  $240_{10}$  onto the bus. It also asserts two control lines to the M(16 word scratchpad); one of these lines selects the register LINTOT, the second line instructs the module to read data from the bus. Timing interlocks from the K(bus) control module, prevent data from being read from the bus before it has settled. This Ke also asserts control lines to set the two flags; however, this does not involve use of the bus and can be performed in parallel with the previous operation.

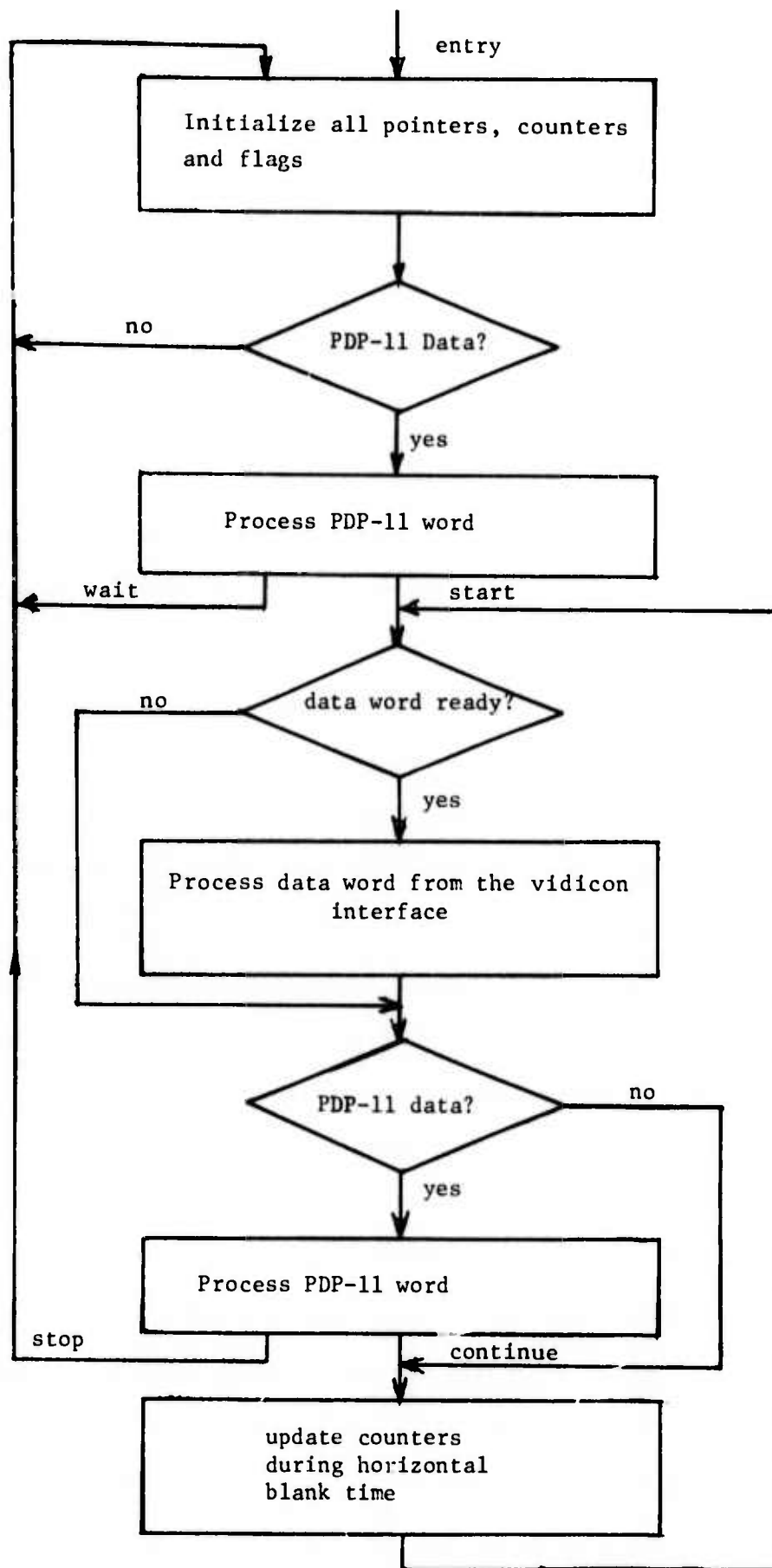
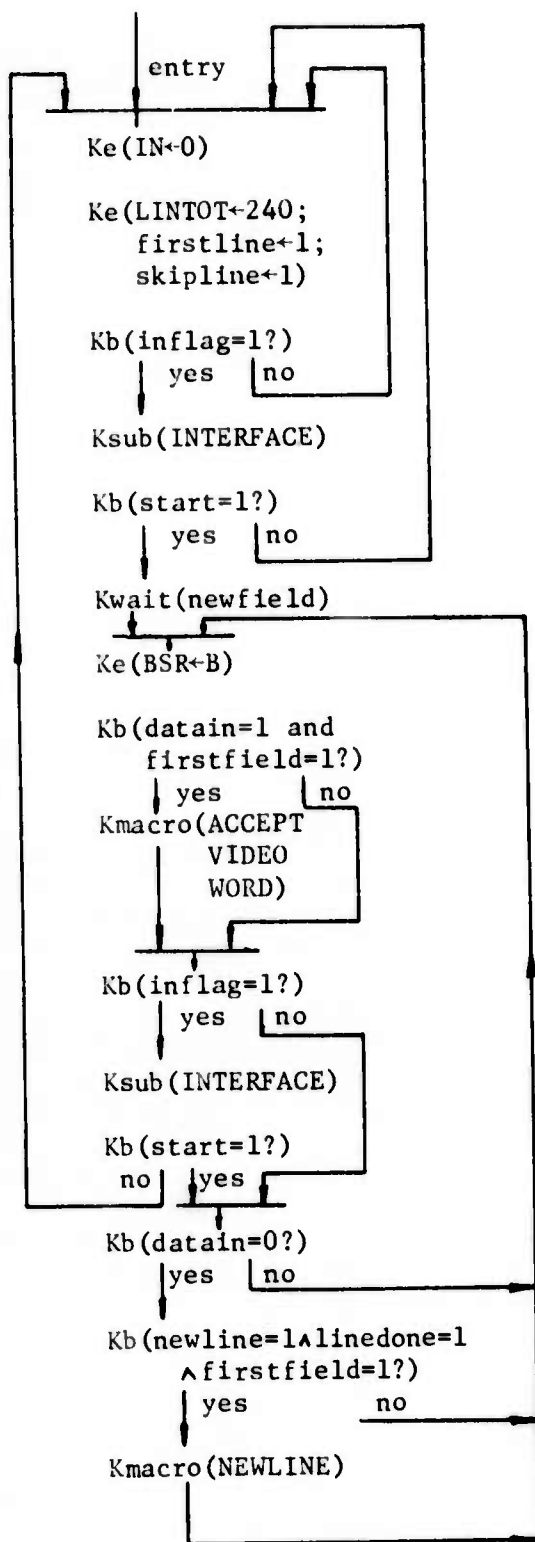


Figure 18. Flow Diagram of Video Processor and Device Interface.



MODULE	REGISTERS AND FLAGS	DESCRIPTION
T(lights and switches)	registers: LIGHTS SWITCHES	displays bus transfers manual data input controls bus use
K(bus control)	register: BSR flags: OVF BSR=0 BSR>0 BSR<0	stores last data on bus overflow flag bus data = 0 flag bus data > 0 flag bus data < 0 flag
DM(general purpose arithmetic unit)	registers: A,B	performs arithmetic and logical operations
M(16 word scratchpad)	registers: IN LINTOT LINE LINEMAX	store data to be operated upon picture element pointer horizontal line counter picture element line counter number of lines in a picture element
M(256 word scratchpad)	register: MA MB	memory address pointer memory buffer
M(4 word, read-only)	registers: MASK 240 <sub>10</sub>	(=017777 <sub>8</sub> ) used to mask code from data number of horizontal scan lines in 1 video field being digitized
DM(boolean flags)	flags: inflag outflag firstline start skipline lastline firstfield linedone	PDP-11 data word ready output word for PDP-11 ready current horizontal line is first line of a picture element start/stop bit for video processor set to skip first horizontal line of video field current horizontal line is last line of picture element used to skip every other video field disables counter updates during horizontal blanking time
M(interface, #1)	registers: INREG1 OUTREG1	interfaces an RTM system to an external system input word from PDP-11 output word to PDP-11
M(interface, #2)	registers: INREG2 OUTREG2	input word from vidicon inter- face output to vidicon interface
T(vidicon interface)	flags: datain newfield newline	data word from vidicon inter- face ready vertical blanking period horizontal blanking period
T(digital-to analog converter, #1,2)	registers: XA<7:0> YA<7:0>	holds data word to be con- verted by D/A #1 holds data to be converted by D/A #2

Table 3. RTM modules used in the Optical/Digital Interface.



Initialize registers and flags

Data word from PDP-11?

Process data word

Start video processor?

Wait for start of new video field

Dummy timing operation

Data from vidicon interface?

Process data word

Data word from PDP-11?

Process PDP-11 data word

Stop video processor

Has horizontal blank period started?

Update counters

Figure 19. Main Processor RTM Design

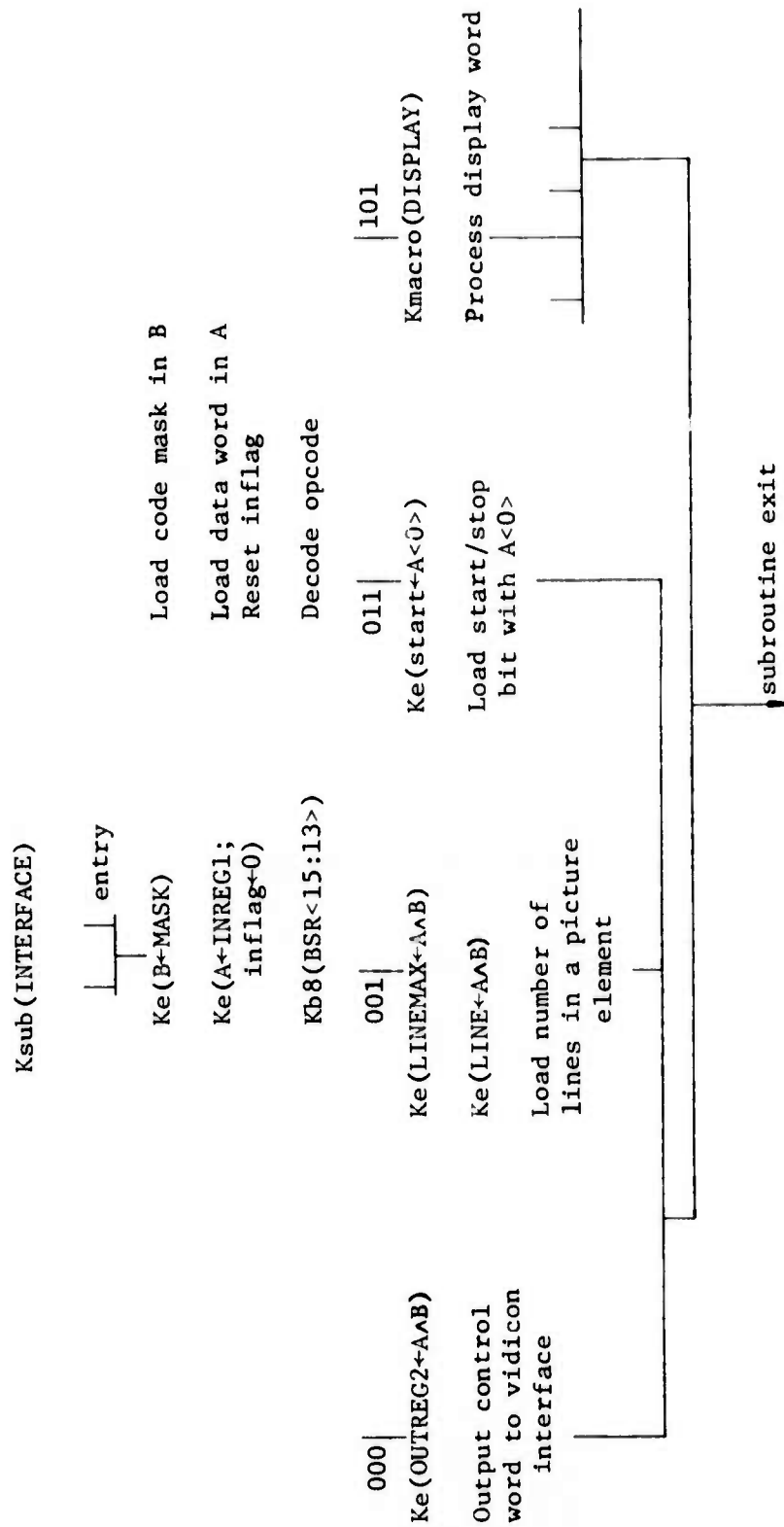
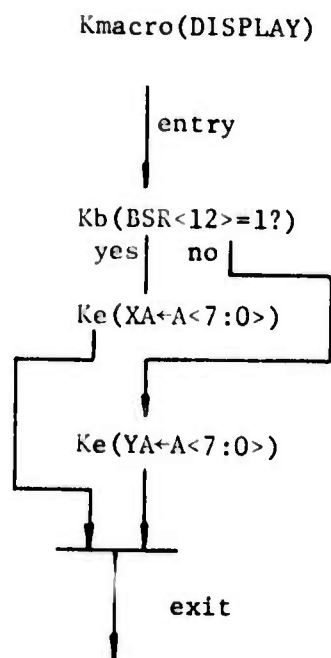


Figure 20. RTM Subroutine INTERFACE



Is this X-deflection or  
Y-deflection data.

X-deflection. Load low  
order byte of A in  
X-deflection D/A.

Y-deflection. Load low  
order byte of A in  
Y-deflection D/A

Figure 21. Kmacro (DISPLAY)

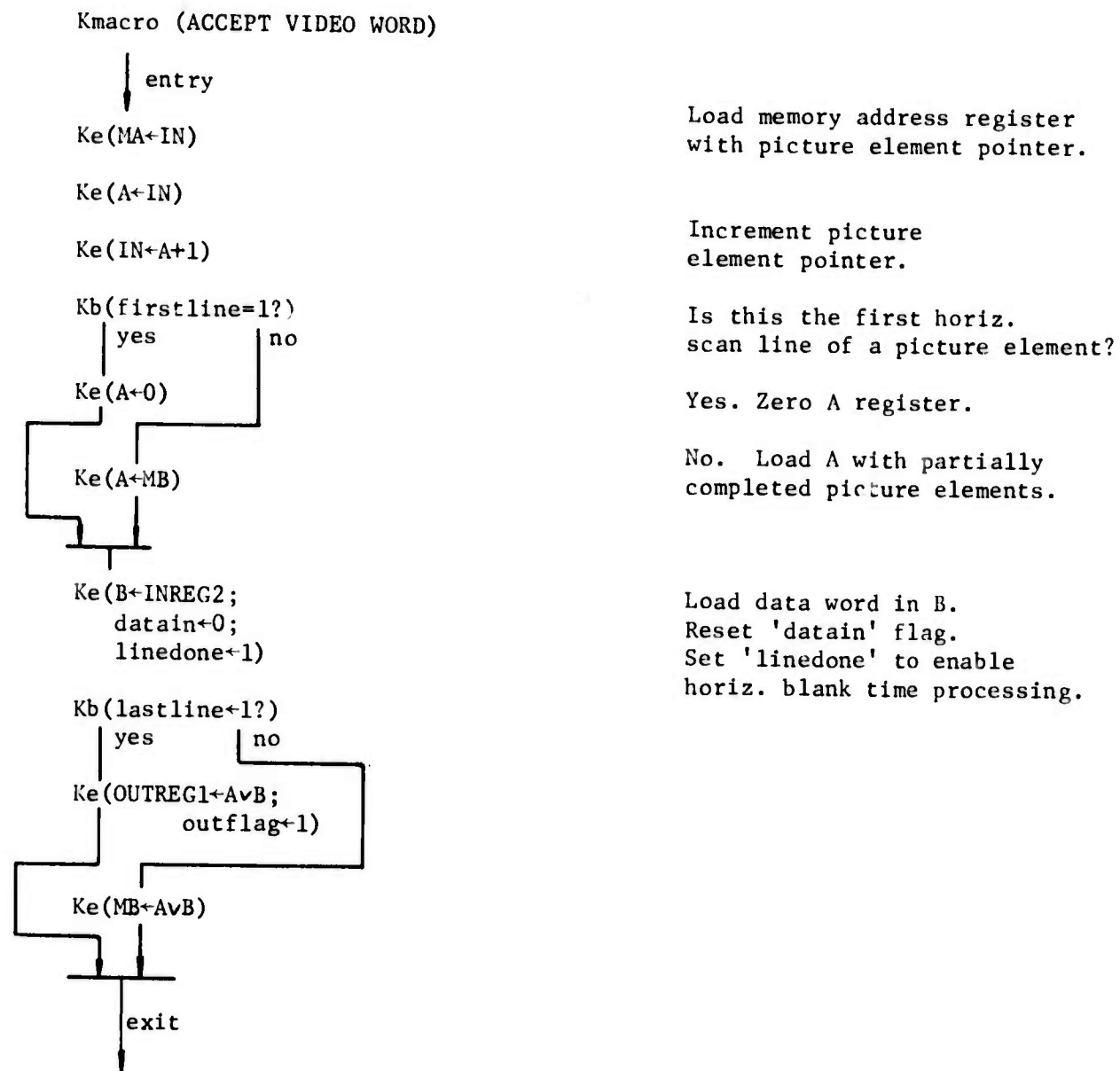


Figure 22. Kmacro (ACCEPT VIDEO WORD)

Kmacro(NEWLINE)

linedone←0

Kb(newfield=1?)

no

yes

B

Ke(IN←0)

Kb(skipline=1?)

no

yes

C

Ke(A←A;  
firstline←0)

Kb(lastline=1?)

yes

no

lastline←0

firstline←1

Ke(A←LINE)

Ke(LINE←A-1)

Kb(LINE=0?)

Yes

no

Ke(A←LINEMAX;  
lastline←1)

Ke(LINE←A)

Ke(A←LINTOT)

Ke(LINTOT←A-1)

Kb(LINTOT=0?)

yes

no

lastline←1

Kb(LINTOT<0?)

yes

no

Ke(A←A;  
lastline←0;  
firstline←1)

Disable further counter  
updates during this  
horiz. blank time.

Is this the vertical blanking  
period?

Reset picture element pointer.

Is this the first horiz.  
scan line?

Dummy operation. Reset  
firstline.

Is this the last line of a  
picture element?

Reset lastline flag. Set  
firstline to indicate next  
line is first line of a  
picture element.

Decrement picture  
element line counter.

Has it reached zero?

Reset picture element line  
counter. Set lastline to  
indicate next line is  
last line of a picture  
element.

Decrement horizontal  
line counter.

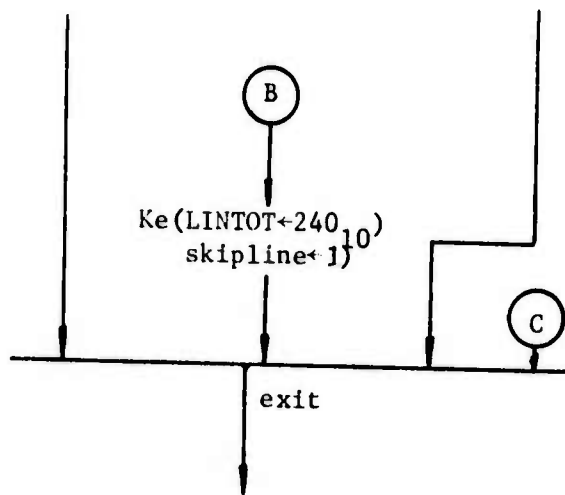
Is it zero?

Set lastline to complete  
current picture element  
with next line.

Is LINTOT<0?

Remaining horizontal lines  
are not included in digitization  
process. Set flags to prevent  
further data transfers to PDP-11.

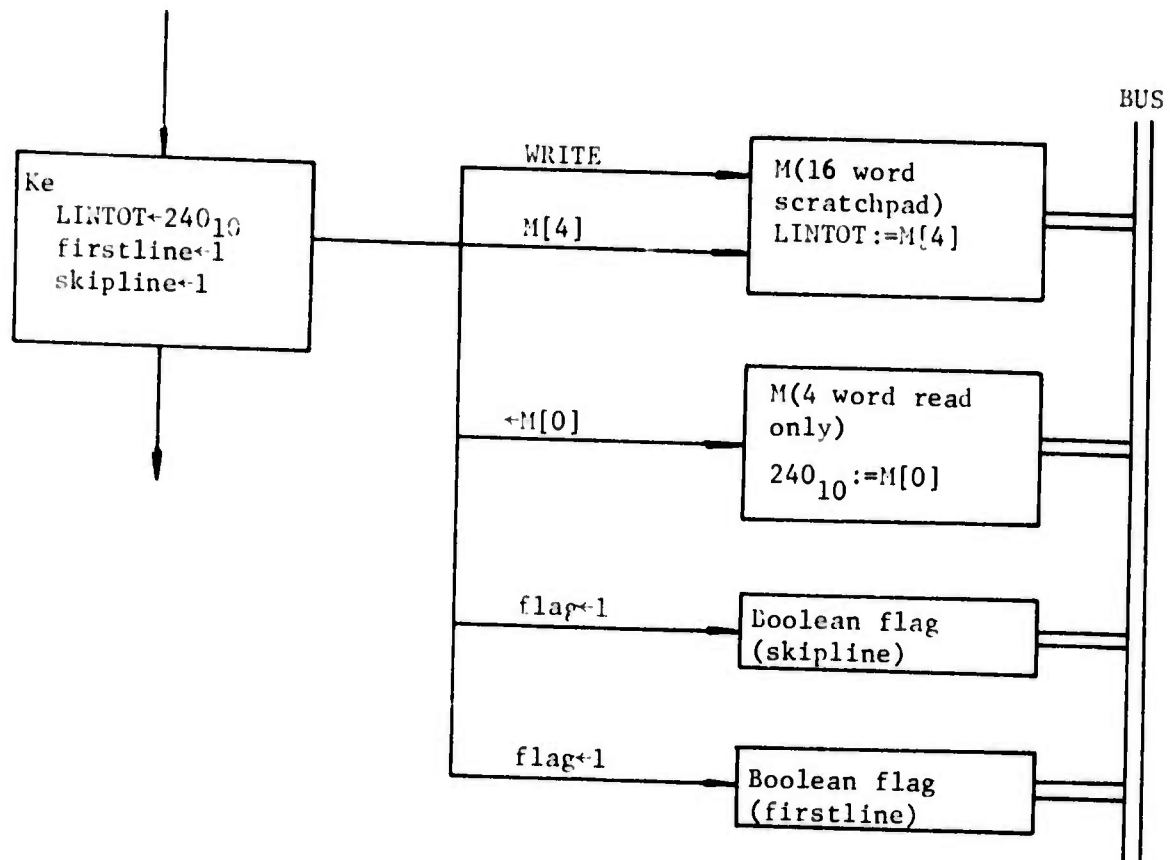
Kmacro (NEWLINE) - continued - 2nd page



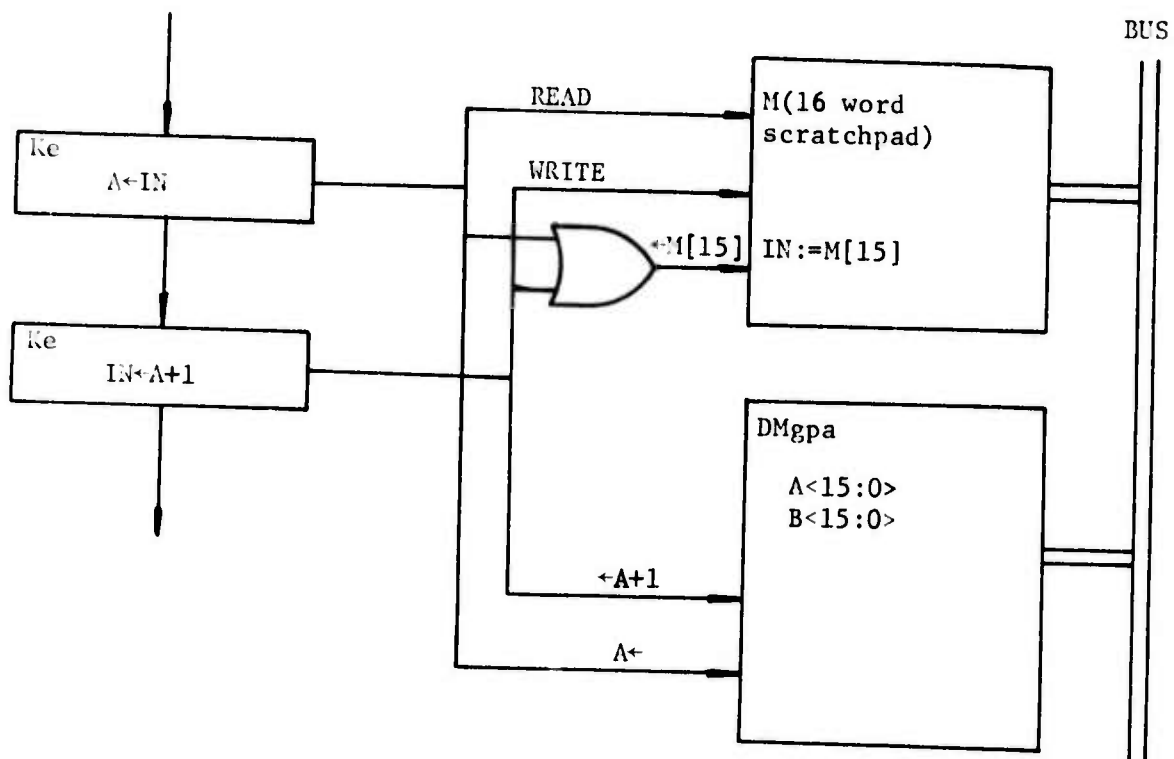
Vertical blanking period (previous frame is complete). Reset horizontal line pointer. Set skip-line to disregard first horizontal line when processing next field.

Figure 23. Kmacro (NEWLINE)





(a)



(b)

Figure 24. Examples of Ke evoke line connections.

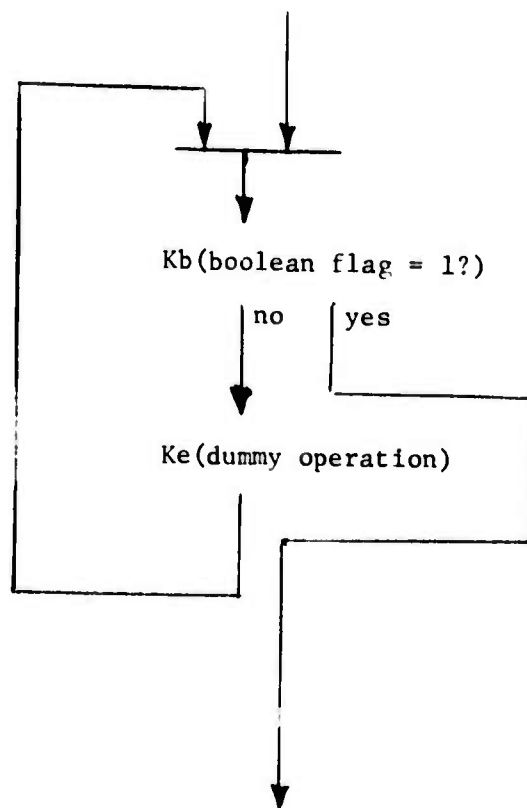


Figure 25. The Kwait module

Figure 24(a) shows the physical configuration of the control lines for this Ke. Note that the Ke's pass control on from one Ke to the next Ke. The operation "IN-0" is completed first; control then passes to the second Ke which initiates the "LINTOT-240" operation; etc.

The Kb's are 2-way branch modules which examine the value of a boolean and choose between two control paths on that basis. The first Kb in Figure 19 examines "inflag"; if it is 1, control passes downward to a Ksub; if its value is 0, control returns to the first Ke. Eight way branch modules called Kb8's also exist (Figure 20). These examine three booleans and choose among eight paths on that basis.

A Ksub, or subroutine module, is used to execute the same series of operations from different parts of the processor without physically duplicating the hardware each time. Its use is analagous to the subroutine or procedure call in high level software languages. In Figure 19, the two control modules Ksub (INTERFACE) transfer control to the INTERFACE subroutine shown in Figure 20. When the subroutine has finished execution, control returns to the Ksub which called it, and is then passed to the next control module.

The Kwait module does not physically exist but represents the configuration, shown in Figure 25; it merely prevents control from passing on until the designated boolean attains a value of 1. In Figure 19, Kwait(newfield) inhibits video processing until the beginning of a new field ("newfield" is the positive vertical blanking signal).

A Kmacro is simply a logical designation of a series of control steps, much like the Kwait, except that it is completely general. It is again not a physical module but is used in RTM documentation to allow the RTM diagrams to be broken into logical components. In the physical implementation of an RTM processor, a Kmacro is replaced by the control steps it represents. This corresponds to the use of a macro call in some software assemblers, or to the "in place" function calls in higher level languages. The actual control steps represented by Kmacro(ACCEPT VIDEO WORD) and Kmacro(NEWLINE) are shown in Figures 22 and 23, respectively.

Control from several paths can be merged into one control path using a serial merge gate. Such a gate is labeled at the top of Figure 19, and is used throughout the RTM diagrams.

Figure 24(b) shows an additional example of the physical configuration of the Ke control lines. The example chosen is incrementing the contents of register IN by utilizing the DMgpa. This corresponds to the second and third evokes in Kmacro(ACCEPT VIDEO WORD) of Figure 22.

#### 4.3. PDP-11 Interface Control

Figures 26 and 27 show the digital circuits designed to control the PDP-11 general purpose interface (DR11-A) and DMA interface (DR11-B), respectively. The DR11-A control allows for one word data transfers between the PDP-11 and an RTM system, with processor intervention required. The maximum data transfer rate from the video processor to the PDP-11 using the DR11-A is about one word every 5  $\mu$ sec.

The DR11-B control also allows one word data transfers from the PDP-11 to the device interface with processor intervention. Block transfers to or from the PDP-11 can also be initiated in DMA mode. Provisions are included to detect attempts to transfer a word to the PDP-11 in DMA mode, before the previous word has been accepted. The decoding of the FNCT 1,2,3 bits in the DR11-B is defined below:

FNCT1	FNCT2	FNCT3	MODE
0	0	0	PDP-11 TO INTERFACE, BLOCK TRANSFER
1	0	0	INTERFACE TO PDP-11, BLOCK TRANSFER
0	0	1	PDP-11 TO INTERFACE, SINGLE WORD TRANSFER

Figure 28 shows the signal connections between the DR11-B and the optical/digital interface. Note that the FNCT1 bit is tied directly to the C1 control bit of the DR11-B [18].

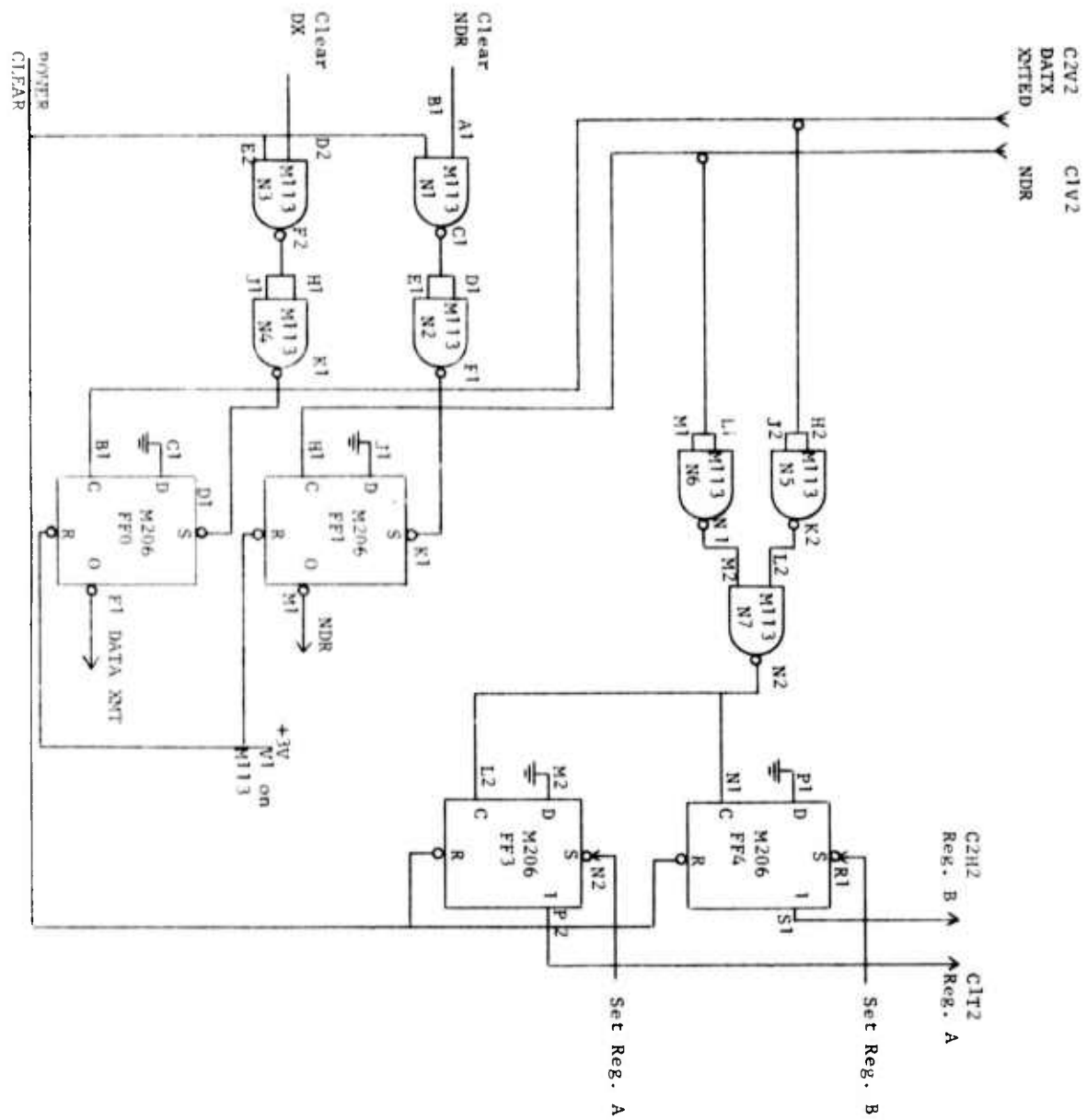


Figure 26. DR11-A Digital control

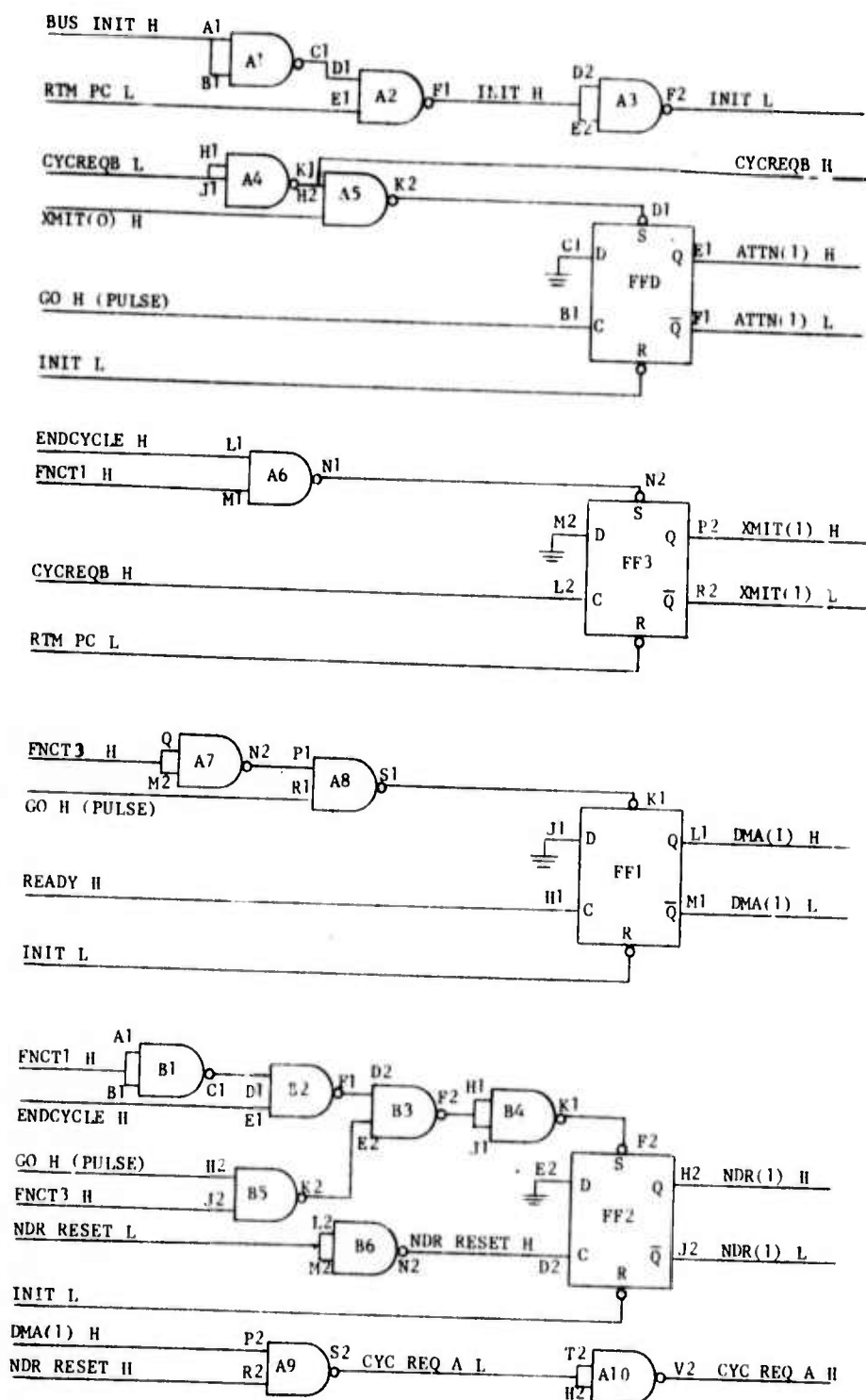


Figure 27. DR11-B digital control

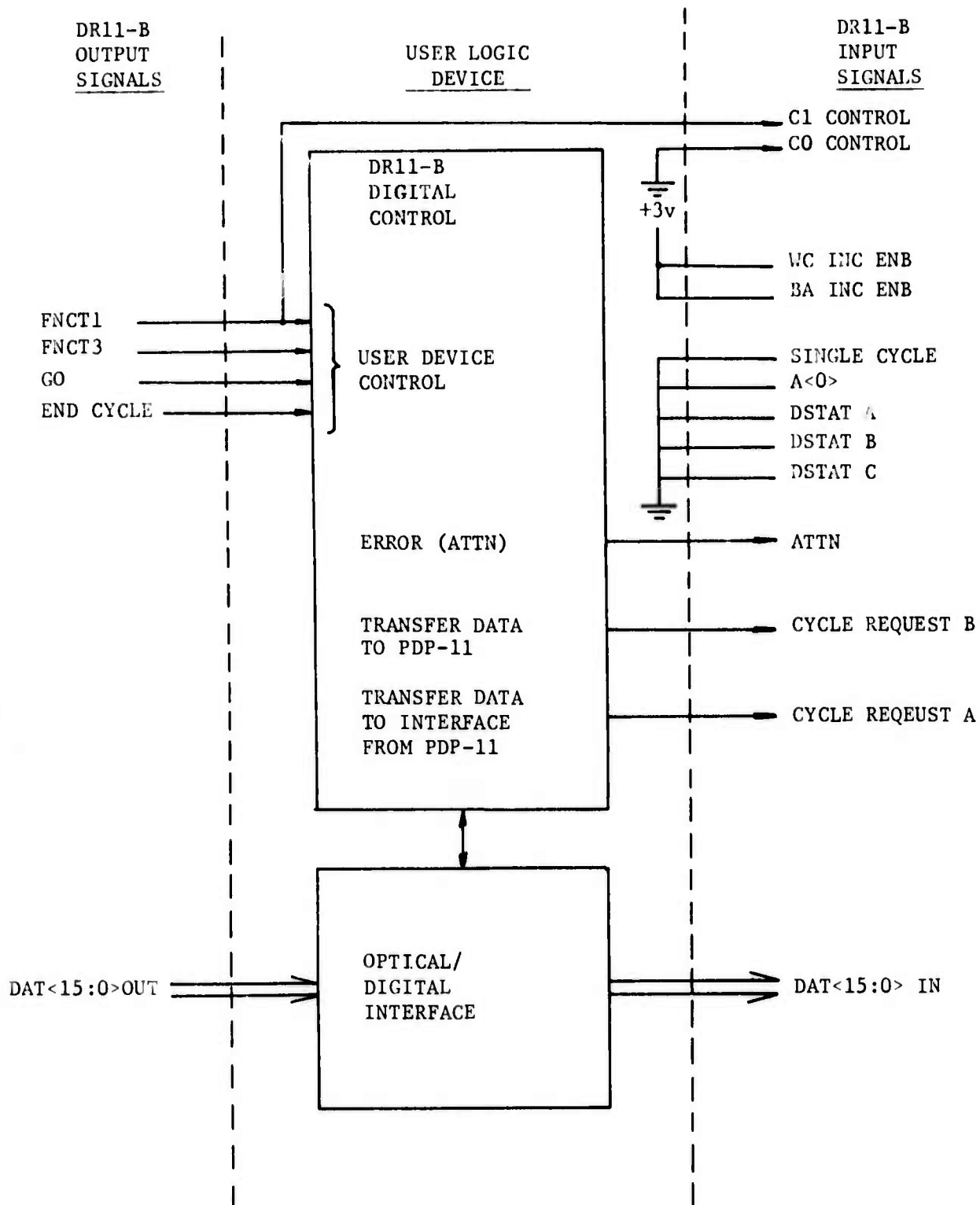


Figure 28. Block diagram of DR11-B control signal connections.



## 5. EXPERIMENTAL RESULTS

The digital portion of the hybrid processor, as it currently exists, is shown in Figure 29. The oscilloscope at the left is used as the CRT display. The optical/digital interface itself is mounted in the table rack.



Figure 29. Photograph of digital portion of the hybrid processor. From left to right: CRT display, TV monitor, rack mounted optical/digital interface, PDP-11 computer (in floor rack).

Figures 30 and 31 show several digital images as they appear on the CRT display. Each image has been derived from a single TV frame on-line. Figure 30 demonstrates  $64 \times 240$  resolution, the maximum resolution of the system using

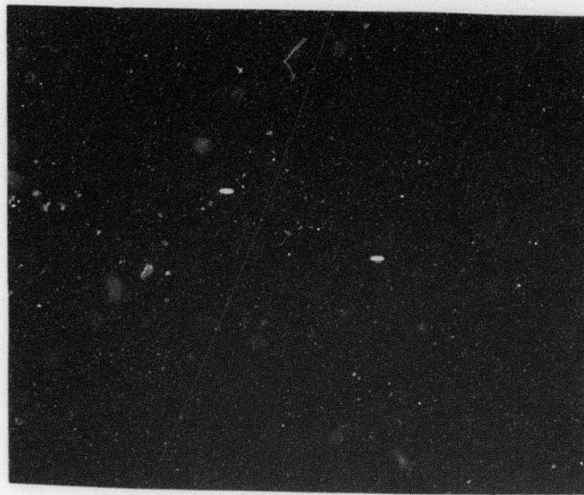
the DR11-A interface. At this time, implementation of the DR11-B is not complete. Such images normally would not be processed by the system, rather it is intended to be used in analyzing correlation and Fourier transform plane patterns.

Figure 31 corresponds to Figure 4 and shows the results of digitizing various optical processor output planes. Figure 31(a) shows the location of the two off-axis spectral components associated with phased array radar processing (Figure 4(b)). Figure 31(b) shows detection of the correlation peak of Figure 4(e). Finally, Figure 31(c) shows detection of the correlation peak corresponding to the word 'HIGH' in the input plane pictured in Figure 4(f).

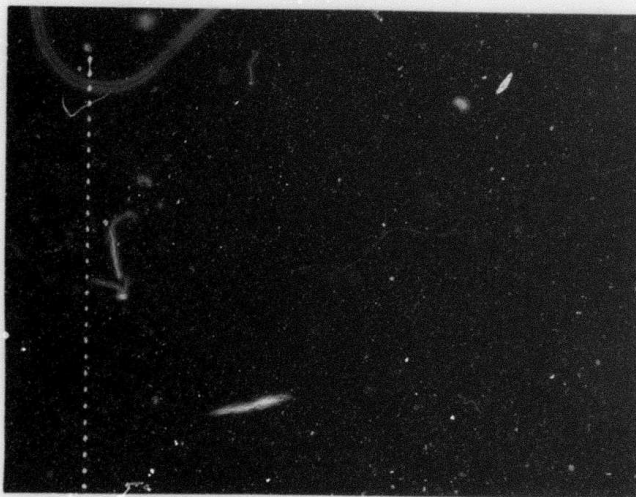


Figure 30. CRT display image demonstrating digital resolution of 64 x 240 picture elements for one on-line TV frame.

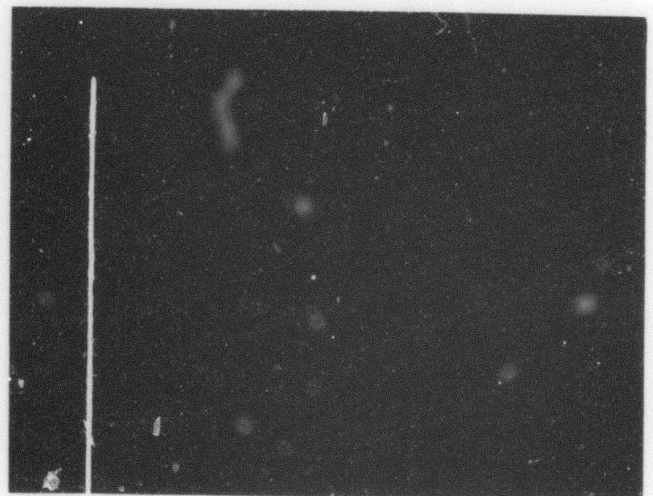




31(a). Off-axis spectral components for phased array radar processing output of 4(b). resolution: 64 x 240.



31(b). Correlation peak of scene correlation output of 4(e). Resolution: 32 x 30.



31(c). Correlation peak of text correlation output of 4(g). Resolution: 64 x 120.

Figure 31. Binary digital images of the optical processor output planes shown in Figure 4. Ignore vertical lines at sides of the pictures.

Current research efforts, beyond these initial demonstrations and documentations, are concentrating on assessing the interactive nature of the hybrid processor and the resolution requirements of the system in a wide variety of processing modes. Specific application areas being considered are:

1. Radar processing
2. Edge detection
3. Feature extraction
4. Global and local transforms
5. Optical subtraction
6. Optical correlation
7. Contour mapping
8. Stereo-pair imaging

A future report and the Ph.D. thesis of Warren Sterling will report on these and related interactive optical/digital operations of the system.

The complete interaction of the input electron beam addressed modulator, a spatial filter and the output plane digitizer and display has been demonstrated. As the data processing requirements are crystallized the feedback control using the results of the analysis of the output plane data to determine the system's new input plane and spatial filter plane data will be our major research direction. This feedback mode will probably be used in a radar processing application for fine range resolution, etc.

Improvements currently being added to the optical/digital interface involve the design and implementation of comprehensive software and include:

1. conversational (interpretive) and stored object code (compiled)  
programming modes

2. a high level programming language with arithmetic capability consistent with image processing requirements
3. control features for various peripheral devices in real-time processing
4. common image processing functions implemented as system procedures
5. extensibility in areas of device control and image processing procedures.

#### SUMMARY

The design of optical/digital interface for a hybrid processor has been presented. This device allows proper allocation of data processing functions to the high data rate optical processor and the flexible digital computer. This results in a computer which combines the best features of optical and digital technologies. The choice of the register transfer level of design, rather than the conventional gate level, resulted in a much shorter implementation time; furthermore, the modularity of such a design permits an in-depth view of the interface which is easily understood. The control logic is similar in structure to software algorithms but attains processing speeds normally associated with hardwired algorithms. This allows for the large amount of data reduction processing necessary to efficiently link an optical processor to a digital computer. In addition, the interface provides the optical processor control capability necessary for its operation in a real-time mode.

#### ACKNOWLEDGMENTS

The authors would like to express their gratitude to William Keicher for constructing the Pockels Tube, Frank Caimi for the design and construction of the tube's electron optics, Dorothy Josephson for typing the manuscript, Bunny Kostkas for drawing the figures, the Office of Naval Research for support on NR-048-600 and the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-73-C-0074), monitored by the Air Force Office of Scientific Research, for providing the RTM modules used in implementing the computer interface.

## References

1. C. G. Bell, A. Newell, Computer Structures: Readings and Examples, McGraw-Hill, New York, 1971.
2. PDP-16 Computer Designers Handbook, Digital Equipment Corporation, Maynard, Mass., 1971.
3. D. Casasent, "A Hybrid Digital/Optical Computer System," IEEE Transactions on Computers, Vol. C-22, No. 9, September, 1973.
4. K. Preston, Jr., "Chapter 4: Use of the Fourier Transformable Properties of Lenses for Signal Spectrum Analysis," Optical and Electro-Optical Information Processing (J. T. Tippett, Ed.), MIT Press, Mass., 1965, pp. 59-68.
5. R. J. Collier, C. B. Burckhardt, L. H. Lin, Optical Holography, Academic Press, New York, 1971.
6. J. W. Goodman, Introduction to Fourier Optics, McGraw-Hill, New York, 1968.
7. G. W. Stroke, An Introduction to Coherent Optics and Holography, second edition, Academic Press, New York, 1969.
8. W. L. Poppelbaum, M. Faiman, D. Casasent, D. Sand, "On-Line Fourier Transform of Video Images," Proc. IEEE, Vol. 56, No. 10, October, 1968, p. 744.
9. D. Casasent, W. Keicher, "An Electro-optic 2-D Spatial Light Modulator for Real-Time ODP," Proc. Electro-Optical Systems Design Conference, New York, 1972, pp. 99-106.
10. G. Marie, "Un Nouveau Despositif de Restitution D'Images Utilisant un Effet Electro-optique: Le Tube Titus," Philips Research Reports, Vol. 22, 1967, pp. 110-132.
11. A. Yariv, Quantum Electronics, Wiley, Inc., New York, 1967.
12. D. Casasent and J. Stephenson, "Electro-Optical Processing of Phased Array Antenna Data," Applied Optics, Vol. 11, No. 5, May, 1972, p. 1269.
13. F. Casasayas, On-line Electro-Optic Processing of Radar Signals (tentative), Ph.D. Dissertation, Carnegie-Mellon University, 1973.
14. C. G. Bell, J. Grason, "The Register Transfer Module Design Concept," Computer Design, May, 1971, pp. 87-94.
15. C. G. Bell, J. L. Eggert, J. Grason, P. Williams, "The Description and Use of Register Transfer Modules (RTM's)," IEEE Transactions on Computers, Vol. C-21, No. 5, May, 1972, pp. 495-500.
16. C. G. Bell, J. Grason, A. Newell, Designing Computers and Digital Systems Using PDP-16 Register Transfer Modules, Digital Press, Maynard, Mass., 1972.



17. D. Casasent, W. Sterling, "A Register Transfer Module FFT Processor for Speech Analysis," Proc. Fall Joint Computer Conference, 1972, Vol. 41, pp. 709-717.
18. DR11-B Manual, Digital Equipment Corporation, Maynard, Mass., 1973.

## 8. APPENDIX A1: INSTRUCTION SET PROCESSOR (ISP) DESCRIPTION

ISP is a notation for defining the action of a digital processor or computer, at the register transfer level. A complete explanation of ISP can be found in reference [1], which contains the ISP of 14 computers. For completeness, the ISP description of the optical/digital interface is presented here.

### ISP OF OPTICAL/DIGITAL INTERFACE

#### Console State

SWITCHES<15:0>	data entry
LIGHTS<15:0>	data display

keys for Start, Power Clear, Single Step not included.

#### Processor State

BSR<15:0>	bus sense register
A<15:0>	arithmetic unit register
B<15:0>	arithmetic unit register
BSR = 0	bus data = 0 flag
BSR < 0	bus data < 0 flag
firstline	internal flags defined in Table 3
start	
skipline	
lastline	
firstfield	
linedone	
newfield	
newline	
op := BSR<15:13>	
xaxis := BSR<12>	
	control bits
	CRT data flag

#### I/O State

INREG1<15:0>	input data from PDP-11
OUTREG1<15:0>	output data to PDP-11
INREG2<15:0>	input from vidicon interface
OUTREG2<15:0>	output to vidicon interface
XA<7:0>	CRT X-deflection register
YA<7:0>	CRT Y-deflection register
inflag	input from PDP-11 ready
outflag	output for PDP-11 ready

## M State

```

M[0:377]<15:0>
M(16)[0:17]8<15:0>
M(rom)[0:3]8<15:0>
LINE      := M(16)[1]
LINEMAX   := M(16)[2]
LINTOT    := M(16)[4]
IN        := M(16)[17]8
24010     := M(rom)[0]
MASK      := M(rom)[1]

```

256 word scratchpad memory  
 16 word scratchpad memory  
 4 word read-only memory  
 Internal registers defined  
 in Table 3

= 240<sub>10</sub>  
 = 017777<sub>8</sub>

Execution\_process := (

```

  IN ← 0;
  LINTOT ← 24010;
  firstline ← skipline ← 1;
  (inflag = 1) →
    Instruction_execution;
  (start = 1) → (
    Wait_for_newfield;
    Digitization_process
  )
  Execution_process
)

```

Initialize counters and flags

Process any data from PDP-11

Enter digitization process at  
beginning of new field

Repeat main process

Instruction\_execution := (

```

  B ← MASK;
  A ← INREG1;
  inflag ← 0;
  (op = 0) → OUTREG2 ← A^B;
  (op = 1) →
    LINEMAX ← LINE ← A^B;
  (op = 3) → start ← A<0>
  (op = 5) → Display_process
)

```

Load control mask  
 Load input data  
 Reset input flag  
 Output data to vidicon interface  
 Initialize horizontal line counters

Set start bit  
 Process CRT data

Wait\_for\_newfield :=

```

  (newfield = 0) → Wait_for_newfield;

```

Display\_process :=

```

  (xaxis = 1) → XA ← A<7:0>
  else        YA ← A<7:0>

```

Load CRT data word in X or Y  
 deflection register

Digitization\_process := (

```
(datain = 1 ∧ firstfield = 1) →
  Accept_video;
(inflag = 1) → (
  Instruction_execution;
  (start = 0) →
    Execution_process
  );
(datain = 0 ∧ newline = 1 ∧
  linedone = 1 ∧ firstfield = 1) →
  Update_pointers;
Digitization_process
)
```

Process data from vidicon interface

Process any data from PDP-11

If start bit reset, repeat main process

Update counters during horizontal blanking period

Loop in this process

Accept\_video := (

```
(firstline = 1) → A ← 0
else
  A ← M[IN];
B ← INREG2;
datain ← 0;
linedone ← 1;

(lastline = 1) → (
  OUTREG1 ← AVB;
  outflag ← 1
)
else
  M[IN] ← AVB;
  IN ← IN + 1
)
```

Load A with partially completed digital elements

Load vidicon interface data in B

Reset data ready flag

Enable execution of Update\_pointers process during next horizontal blanking period

If elements complete, output to PDP-11

Replace partial elements in memory

Update memory pointer

Update\_pointers := (

```
linedone ← 0;

(newfield = 1) → (
  LINTOT ← 240;
  skipline ← 1
)
else
  IN ← 0;
  (skipline = 0) → (
    firstline ← 0;
    (lastline = 1) → (
      lastline ← 0;
      firstline ← 1
    );
  );
```

Disable re-execution of this process during current horizontal blanking period

If vertical blanking period, reset counter and flag

Reset memory pointer

Reset flag

If elements complete, reset flags to start a new set of elements

```

LINE ← LINE-1;
(LINE = 0) → (
    LINE ← LINEMAX;
    lastline ← 1
);
LINTOT ← LINTOT-1;
(LINTOT = 0) →
    lastline ← 1;
(LINTOT < 0) → (
    lastline ← 0;
    firstline ← 1
    )
    )
    ).

```

Decrement element line pointer

Next horizontal line will be last line of element

Decrement horizontal line pointer

Next line is last horizontal line, and must be last line of element

Additional lines not included in digital picture

Flags set to ignore them